OpenVMS System Management Techniques, Tools, & Tricks

F. Arthur Cochrane

Senior System Admin
CSX Technology

Outline

- DCL Techniques
- Network Topics
- SYSMAN
- System Information Command Procedures
- Using Logical Names
- AMDS
- OpenVMS Management Tools

Disclaimer

- I am going to cover a lot of various topics in the seminar
- Some you may know and some you may not
- Some you may like and some you may think I am crazy
- If you decide not to use any of the ideas presented at least I hope you learn something about VMS you did not know before entering the room
- Maybe someday you will have an occasion to use one of these ideas to make your VMS system easier to maintain

DCL Techniques

- DCL Control Key Line Editing
- F$ELEMENT
- F$PARSE
- F$FAO
- F$ENVIROMENT
- F$MODE
- Foreign Commands

DCL Control Key Line Editing

A	Insert Mode
B	Recall Last Entered Command
C	Cancel Certain Commands
D	Move Cursor One Character to the Left
E	Move the Cursor to the End of the Line
F	Move Cursor One Character to the Right
G	
H	Move the Cursor to the Beginning of the Line (BACKSPACE)
I	Horizontal TAB
J	Delete the Preceding Word (LINEFEED)
K	
L	
M	Line Terminator (CR)
N	
O	Suppress Output
P	
Q	Resume Display
R	Retype the Line
S	Hold Display
T	Display Information on Process
U	Delete the Line to the Left of the Cursor
V	Activate Function Keys F7-F14 (See DEFINE/KEY)
W	Redisplay the Screen
X	Discard the Current Line
Y	Cancels Entire Operation
Z	Close a File

DCL Uptime Command

- A simple “uptime” command
  - `$ upt*ime :== show system/noprocess`

```
EISNER $ uptime
OpenVMS V7.2-1  on node EISNER  17-JUN-2004 09:59:53.02  Uptime  24 16:02:10
```

- This shows only the banner line of the show system command
- The /noprocess qualifier is new to V6.2

Assignment Statement

- The = verse the := assignment statement
- The = can be for numeric or string assignment

```
$ A = 3
$ B = "abc"
```

  - Note the required quotes for the alpha assignment
- The := is for string assignment only

```
$ C = abc
```

  - NOTE that without quotes case is NOT preserved and leading and trailing spaces are trimmed.
- I use the := for assigning command shortcuts

```
$ upt*ime :== show system/noprocess
```

Bit Assignment for Strings

- variable[start_bit,bit_count]=numeric_value
- Examples:

```
$ ESC[0,8]=27
$ CR[0,8]=13
$ LF[0,8]=10
$ FF[0,8]=12
$ CRLF[0,8]=13
$ CRLF[8,8]=10
```

- Notice the last two show creating a two byte string
- This allows command files to be printed without embedded control characters that can cause strange printouts

VMS is a Great Number Converter

- Remember numbers can be other than decimal

```
$ A = %XFF
$ B = %O177
$ C = 10 + %X10 + %O10
$ show symb/all
  A = 255   Hex = 000000FF  Octal = 00000000377
  B = 127   Hex = 0000007F  Octal = 00000000177
  C = 34   Hex = 00000022  Octal = 00000000042
```

```
SCSSYSTEMID = (10*1024) + 90 ! DECnet address 10.90
```

- The line above is used in AUTOGEN to let VMS do the math to calculate the parameter.

F$ELEMENT

- Technique learned from ‘Writing Real Programs in DCL’ book

```
$ com_file = f$element(0,";",f$environment("PROCEDURE"))
```

- May have seen taking the file and subtracting the version number with F$PARSE

```
F$PARSE(file,,,”VERSION”,”SYNTAX_ONLY”)
Example:
$ file_spec = file_spec –
  F$PARSE(file,,,”VERSION”,”SYNTAX_ONLY”)
```

F$PARSE

- Handy lexical for extracting parts of a file spec

```
F$PARSE(file,,,"NAME","SYNTAX_ONLY")
```

- Handy for filling in the rest of file spec given only part of the file necessary

```
F$PARSE("sys$login:;",file,,,"SYNTAX_ONLY")
F$PARSE("sys_manager:;",f$environment("PROCEDURE"),,,"SYNTAX_ONLY")
```

F$FAO

- F$FAO(control string,argument)
- Great lexical for lining up columns of output

```
DSA100:      PAGE_SWAP     RZ24    409792  360802  88%   48990  11%    1   0
DSA150:      USER3         RZ24    409792  394261  96%   15531   3%    1   0
DSA200:      USER1         RZ24    409792  362743  88%   47049  11%   65   0
DSA300:      ARTHUR_SYS    RZ24    409792  331090  80%   78702  19%  722   0
```

- USE !AS for ASCII Strings and !UL for numbers
- !xAS where x the the number of spaces to use no matter the size of the file

```
$ output=f$fao("!16AS!14* !8AS",device_d,device_name)
```

F$ENVIRONMENT

```
$ com_file = f$environment("PROCEDURE")
```

- This returns the full file spec of the command procedure that the lexical is called from
- This is great for batch jobs when you wish to submit the same file to run again
- Files can be renamed, but will continue to function since the lexical will get the new file name

Batch Resubmit

```
$ submit_time=f$cvtime("today+7-19:00","ABSOLUTE")
$ com_file =
  f$element(0,";",f$environment("PROCEDURE"))
$ log_file = -
  f$parse("log_files:.log",com_file,,,"SYNTAX_ONLY")
$ submit/user=system/after="''submit_time'"-
/noprint/log_file='log_file'/name="Backup Procedure"
  - 'com_file'
```

Updated Batch Jobs in Queue

- How a submitted batch command procedure can execute the most recent version

```
$! Check for update
$ check_file=f$search(file)   ! get latest file on disk name
$ file_running=f$environment("PROCEDURE")    ! this files name
$ if check_file .nes. file_running   ! if not same submit latest
$ then
$       submit/noprinter/keep/notify/name="''name'"-
        /note="''note'"/user=system/log_file='log_file' 'file'
$ exit
$ endif
```

F$MODE

- F$MODE() returns one of four values
  - Interactive
  - Batch
  - Network
  - Other
- Can be used to have one command procedure do different functions depending the mode of activation

F$MODE Usage

```
$ goto `f$mode()'
$!
$interactive:
$ exit
$batch:
$ exit
$network:
$ exit
$other:
$ exit
```

Foreign Commands

- Foreign commands are DCL symbols that allow programs to be run but have parameters and qualifiers passed just like programs with verbs in the DCL tables.
- Examples:

```
$ uaf :== "$authorize"
$ ncp :== "$ncp"
```

Foreign Command Usage

```
$ NCP :== "run sys$system:ncp"
$ NCP :== "$ncp"
```

- The above will do the same thing if it is used alone of the DCL command line.
- However the following will give different results depending on the symbol above used

```
$ ncp show exec char
```

DCL$PATH

- In VMS 6.2 a new logical can be defined to point to a directory of .COM or .EXE utilities
- Normally if a command is entered and there is not a DCL verb or command symbol defined then DCL reports
  - `DCL-W-IVVERB, unrecognized command verb - check validity and spelling`
- The DCL$PATH logical causes DCL to look for a .COM or .EXE with a file name of the invalid verb entered
- This behavior is similar to the PATH option found in DOS

Pre 6.2 Method

- The same thing can be done in pre v6.2 with a command procedure
- Search a utility directory for .COM and .EXE files and define a foreign command for each found
  - Note search for .COM files and then .EXE files
  - If a filename exists as a .COM and .EXE the .COM file is defined as the foreign command
  - This allows a command procedure to be used to set up a executable program

One Difference in Methods

- With the DCL$PATH method any new utility file added to the directory will be able to be used as DCL will search for each unrecognized verb each time one entered
- With the command procedure commands are defined only once

Startup Feedback

- I like to know what a system is doing when it boots
- So I have the following line in all my startup command procedures

```
$ set noon
$ save_verify = f$verify(p2)
$ file=f$environment("procedure")
$ name=f$parse(file,,,"NAME","SYNTAX_ONLY")
$ write sys$output f$fao("%STARTUP-I-!AS, Startup executing !AS",name,file)
$!
```

- P2 for the VMS verify startup option

Utility Initialization Files

- Mail
  - mail$edit
    - callable_tpu
  - mail$init
- EVE/TPU
  - eve$init
  - tpu$command
  - tpu$section
- SYSMAN
  - sysmanini
- Show Cluster
  - show_cluster$init
- Analysis
  - sda$inits
- Debug
  - dbg$init

Utility Convention

- If a utility allows defining keys I try to do the following in each utility
  - F20 is Exit and Gold F20 is Quit
  - F19 set the screen to 80 columns & Gold F19 132 columns
  - Help does Help
- In DCL my define keys are similar
  - Help is Help & Gold Help displays HELP & waits for input
  - Do runs Mail
  - F19 sets the terminal to 80 columns & Gold F19 sets the terminal to 132 columns
  - F20 does a RECALL/ALL & Gold F20 display RECALL and waits for characters to match against a previous command

One File Executes Itself

- On DECUServe the user quota is low so instead of having several command files that use 1 or 2 blocks but allocate 3 blocks I have the command files combined it to LOGIN.COM
- LOGIN.COM then defines symbols to execute the subroutines

```
$extract:==@sys$login:login call extract_notes
$disk_snoopy:==@sys$login:login call disk_snoopy
$bb-no:==@sys$login:login gosub bb-no
$eve_files:==@sys$login:login call eve_files
$mail_compress:==@sys$login:login gosub mail_compress
$ti:==@sys$login:login gosub big_time
```

Login Setup

- LOGIN.COM has code at the beginning to check P1 and if null execute normally
- If P1 is not null then it does what is in the parameters (CALL or GOSUB)

```
$if p1 .nes. ""
$then
$'p1' 'p2' 'p3' 'p4' 'p5' 'p6' 'p7' 'p8'
$exit
$endif
```

- This is what VMSINSTAL does
- So my LOGIN.COM is a bunch of subroutines that get called when I login to the system

Maintain one file

- Instead of several files to maintain I like to have subroutines in my startup files
- To be able to access them I have the following lines at the beginning
- This is very good for SYLOGICALS.COM
  - Add a subroutine for some logicals
  - Then call SYLOGICALS with P1 and P2 for the routine
- Other system managers like to have many small command files but I find they get lost in the forest of files in the startup directory

Start of SYLOGICALS.COM

```
$ set noon
$ if p1 .eqs. "" then p1:=full
$ if p1 .nes. "FULL" then show symbol p%
$ gosub 'p1'
$ exit
$!
$call:
$ call "'p2'" "'p3'" "'p4'" "'p5'" "'p6'" "'p7'" "'p8'"
$ return
$!!
$gosub:
$ gosub 'p2'
$ return
$!!
$full:
$minimum:
$min:
$upgrade:
$ save_verify = f$verify(p2 .or. p8)
$ file=f$environment("PROCEDURE")
$ name=f$parse(file,,,"NAME","SYNTAX_ONLY")
$ write sys$output f$fao("%STARTUP-I-!AS, Startup executing !AS",name,file)
$ set noon
```

Defining Group Logicals at Startup

- Desire to define some group logicals for a UIC group
- Can not just DEFINE/TABLE=LNM$GROUP_000250 because the table does not exist for the group until a member of the group executes
- Use a command procedure to run under the group UIC to define the group logicals
- Note: In this example I have a group ID defined in authorize

Group Logical Startup

```
$ set noon
$ file=f$element(0,";",f$environment("PROCEDURE"))
$ group_file=f$parse("decnet_group_logicals.com",file,,,"SYNTAX_ONLY")
$ if f$search(group_file) .nes. ""
$ then
$ group:=decnet
$ run sys$system:loginout.exe -
        /input='group_file' -
        /output=nla0: -
        /error=nla0: -
        /noauthorize -
        /noaccounting -
        /privileges=(grpnam,sysnam) -
        /process_name="Group Logicals" -
        /uic=['group',0]
$ endif
$ exit
```

Example Group Logical File

```
$! Define in DECnet group logical name table for FAL$SERVER if
  person access with old logical x_DISK:
$ define/group/nolog   decus_cd4_disk        decus_cd4:
$ define/group/nolog   decus_cd5_disk        decus_cd5:
$ define/group/nolog   decus_cd6_disk        decus_cd6:
$ define/group/nolog   decus_cd7_disk        decus_cd7:
$ define/group/nolog   decus_cd8_disk        decus_cd8:
$ define/group/nolog   decus_cd9_disk        decus_cd9:
$ define/group/nolog   decus_cd10_disk       decus_cd10:
$ define/group/nolog   decus_cd11_disk       decus_cd11:
$ define/group/nolog   decus_cd12_disk       decus_cd13:
$ define/group/nolog   decus_cd13_disk       decus_cd13:
$ define/group/nolog   decus_cd14_disk       decus_cd14:
$ exit
```

Another Way for Group Logicals

```
$ RUN SYS$SYSTEM:LOGINOUT.EXE/UIC=[300,1]-
                              /INPUT=NL:/OUTPUT=NL:
```

- The UIC specified does not have to exist in the UAF
- The example creates the LNM$GROUP_000300 logical name table
- Logical names can now be created in the table

SET_NEWUSER.COM

```
$ save_verify=f$verify(0)
$ save_priv=f$setprv("ALL")
$ uaf:=$authorize
$ if p1 .eqs. "" then inquire p1 "User Badge to set Password back to newuser"
$ command=f$fao("UAF MODIFY !AS/PASSWORD=NEWUSER/FLAG=(NOPWD_EXPIRED,NODISUSER)/PWDEXPIRED",p1)
$ write sys$output f$fao("$ !AS",command)
$ 'command'
$ temp=f$setprv(save_priv)
$ temp=f$verify(save_verify)
$ exit
```

- I can never remember the qualifiers for setting an account for a first time login with a forced password change

Autogen Callout

- After AUTOGEN.COM does the parameter settings the setparams phase calls SYS$UPDATE:AGEN$MAIL.COM
  - If the command procedure exists
  - Note AUTOGEN creates logicals for the phases used
    - agen$p1, agen$p2, and agen$p3

```
$ file=f$environment("PROCEDURE")
$ name=f$parse(file,,,"NAME","SYNTAX_ONLY")
$ write sys$output f$fao("%AUTOGEN-I-!AS, Autogen executing !AS",name,file)
$ subject= -
           f$fao("Autogen feedback report for node !AS at !%D with phases !AS - !AS - !AS",-
           f$getsyi("NODENAME"),0,f$trnlnm("agen$p1"),f$trnlnm("agen$p2"),f$trnlnm("agen$p3"))
$ file:=sys$specific:[sysexe]agen$params.report
$ who_gets:=system
$ mail/subject="'''subject'" 'file' 'who_gets'
$ exit
```

OpenVMS *Feature*

- STARTUP$INTERACTIVE_LOGINS global symbol in SYSTARTUP_VMS.COM sets interactive login limit from default of 64
- If this system is set to zero in SYSTARTUP_VMS.COM then logins are disabled for EVERYONE, even users at the console with OPCOM
- Fix is in SYSTARTUP_VMS.COM

```
$ SET LOGINS/INTERACTIVE=1
$ STARTUP$INTERACTIVE_LOGINS==0
```

VMSINSTAL Parameters

- Parameters for VMSinstal
  - P1 is the product to install
  - P2 is the location to install from
  - P3 is OPTIONS
  - P4 are options
    - A is for using or creating an autoanswer file
    - L is for logging for DCL command VMSinstal executes
    - N is for display/print release notes

SET WATCH

- SET WATCH is an unsupported command which reports information about the files you access.
- It gives you information about the XQP access to files
- This can help see what files a program is accessing and with what type of access
  - The problem could be that a file you did not know the program was accessing has protection set incorrectly
- Example:
  - `$ SET WATCH FILE/CLASS=DIRECTORY`

Condist Menu Program

- The CDMENU Utility provides a menu interface that helps you perform the following tasks easily
  - Display a master index of all products
  - Display new and updated products
  - Fetch or copy release notes
  - Display & print installation guides and other documentation
  - Install products

To Use the Program

- Logical CD$DIRECTORY points to the CDROM directory on one of the CONDIST CDs

```
$ "CD$DIRECTORY" = "CD_AXPLIB1:[CDROM]
```

- The run the cdmenu program

```
$ CDMENU == "$CD$DIRECTORY:CDMENU"
```

- Can define the logical CD_EDIT to indicate the editor to use while displaying documentation on your screen

```
$ "CD_EDIT" = "LSE"
```

DECwindows Tips

- OPCOM Window
- VUE command modifications
- VUE commands

OPCOM Window

- Output OPCOM messages into the Message Window of a VMS workstation
- Create a OPCOM DECterm window from a remote node for OPCOM messages

OPCOM to the Message Window Step 1

- Add these lines to SYS$LOGIN:DECW$LOGIN.COM

```
$ set noon
$ temp=f$setprv("ALL")
$ if f$search("sys$login:opcom.com") .nes. "" then -
  spawn/notify/nowait/input=sys$login:opcom.com/process="Opcom"
$ temp=f$setprv(temp)
```

OPCOM to the Message Window Step 2

- SYS$LOGIN:OPCOM.COM

```
$ set process/name="Xopcom"
$ temp=f$setprv("ALL")
$!
$get_device:
$ device=f$getjpi(f$getjpi(0,"OWNER"),"TERMINAL")
$ if device .nes. ""
$ then
$ wait 00:00:30
$ define/user_mode  sys$command         'device'
$ reply/enable
$ else
$ wait 00:00:10
$ goto get_device
$ endif
$!
$ temp=f$setprv("noall,tmpmbx,netmbx")
$!
$ exit
```

Explanation

- The wait loop is for the session manager to get an FTA device, which is the message window
- Need to have the message started on login

OPCOM in a DECterm

- This can be run to create an OPCOM window on a remote node

```
$!
$! Create Opcom Window
$!
$ node=f$getsyi("NODENAME")
$ create/terminal/little_font/window_attributes=(rows=15,columns=132,title="X
  Opcom for ''node'",icon_name="''node'
  Opcom")/noprocess/define_logical=(table=lnm$job,OPCOM_window)
$ allocate/nolog OPCOM_window
$ temp=f$setprv("SECURITY")
$ x=f$trnlnm("opc$opa0_classes")
$ define/user_node  sys$command         OPCOM_window
$ reply/enable=('x')
$ temp=f$setprv(temp)
$ exit
```

Explanation

- This opcom window is great for monitoring the OPCOM output of a remote node that need to managed closely

Some DECwindows command modifications

- VUE$LIBRARY logicals
  - DECW$SYSCOMMON:[VUE$LIBRARY.USER]
  - DECW$SYSCOMMON:[VUE$LIBRARY.SYSTEM]
    - DECW$SYSCOMMON logical
      - SYS$SYSROOT:

VUE$DCL_COMMAND.COM

- Modify to set the DCL command window to a 48 line page
- Put file in the [VUE$LIBRARY.USER] directory
- This will allow it to be executed first

```
$ @sys$common:[vue$library.system]vue$dcl_command 48
$ exit
```

VUE$EDIT.COM

- Modify to give a 48 page edit window
- Put file in the [VUE$LIBRARY.USER] directory
- This will allow it to be executed first

```
$ @sys$common:[vue$library.system]vue$edit "''p1'" 48
$ exit
```

VUE$DECTERM.COM

- Modify to give a larger DECterm with the icon and window title the node name
- Copy the VUE$DECTERM file from the SYSTEM directory to the USER directory
  - This file does not have parameters to pass it
- Modify the following command

```
$ create/terminal/detached
```

- Add qualifiers to it

```
$create/terminal/detached/window_attributes= -
(title='f$getsyi("NODENAME")',icon_name=-
'f$getsyi("NODENAME")')
```

Using DECwindows Motif for OpenVMS

VUE$EXIT_COMMAND_LOOP
VUE$GET_ALL_SELECTIONS
VUE$GET_NEXT_SELECTION
VUE$GET_QUALIFIERS
VUE$GET_SELECTION_COUNT
VUE$GET_SYMBOL
VUE$HIGHLIGHT_UPDATE
VUE$INQUIRE
VUE$INQUIRE_SYMBOL
VUE$POPDOWN
VUE$POPUP
VUE$POPUP_CONFIRM
VUE$POPUP_FOCUS
VUE$POPUP_HELP
VUE$POPUP_MESSAGE
VUE$POPUP_QUALIFIERS
VUE$POPUP_PROGRESS_BOX
VUE$READ
VUE$RESET_SELECTIONS
VUE$SET_DONE_LABEL
VUE$SET_ERROR_STATUS
VUE$SET_OUTPUT_TITLE
VUE$SET_SYMBOL
VUE$SET_TASK_LABEL
VUE$SUPPRESS_FILE_LABEL
VUE$SUPPRESS_OUTPUT_POPUP
VUE$UPDATE_FILEVIEW

SHOW DISPLAY

- SHOW DISPLAY command

```
SADCSA $ show display

        Device:     WSA27:  [super]
        Node:       PPP42.SRS.GOV
        Transport:  TCPIP
        Server:     0
        Screen:     0
```

SHOW DISPLAY Undocumented Qualifier

- Do a SHOW DISPLAY/SYMBOLS
  - The output on the screen is same but the following global symbols are created

```
SADCSA $ show symbol decw$display*
  DECW$DISPLAY_NODE == "PPP42.SRS.GOV"
  DECW$DISPLAY_SCREEN == "0"
  DECW$DISPLAY_SERVER == "0"
  DECW$DISPLAY_TRANSPORT == "TCPIP"
```

- These can then be used in a command procedure if needed

Special Versions

- Version ;0 displays the latest version of a file
- Version ;-0 displays the oldest version of a file

Logical for Banner for Print Jobs

- Logical `PSM$ANNOUNCE` will print the string it is defined instead of the VMS version sting
- Only saw this documented in a figure text in the v4.4 system management manual

GNV - GNU's Not VMS!
GNU-based, UNIX environment for OpenVMS

- GNV is an open source, GNU based, Unix environment for VMS. It is intended to provide the important subset of Unix/Linux/POSIX necessary to port UNIX software to VMS. GNV consists of two parts: a Unix like shell environment and the CTRL Supplemental Library which provides functions typically found on Unix systems, but are missing, incomplete, or incorrect on VMS.
- http://gnv.sourceforge.net/
- Check the OpenVMS web site for the last version of GNV

GNV UNIX Commands

70 commands:

ar	cp	fgrep	ls	rmdir
tee	basename	csplit	find	makes
diff	touch	bash	cut	fmt
md5sum	sed	tr	bzip2	cxx
fold	mkdir	sh	true	cat
date	gcc	mv	sleep	tsort
cc	df	grep	nl	sort
uname	chgrp	diff	head	od
split	unexpand	chmod	diff3	tar
hostname	paste	strip	uniq	chown
egrep	install	pr	sum	wc
cksum	expand	join	ptx	tac
whoami	cmp	expr	less	pwd
tail	comm	false	ln	rm

Pipe Command

- Executes one or more DCL command strings from the same command line. The PIPE command enables you to perform UNIX style command processing, such as command pipelining, input/output redirection, and conditional and background execution.

Without the PIPE command

```
$! Implement the "$ss searchstring" command
$ if p1 .eqs. ""
$ then
$ write sys$output "no searchstring entered, showing system"
$ show system
$ else
$ id = f$getjpi("","PID")
$ show system/output=sys$login:ss_'id'.tmp
$ search/nolog/nonum sys$login:ss_'id'.tmp uptime, pid, 'p1'
$ delete/noconfirm/nolog sys$login:ss_'id'.tmp;*
$ endif
$ exit
```

With the PIPE command

```
SS == "pipe show system |
 search/nolog/nonumbers sys$pipe
 uptime, pid, "
```

- Several line command procedure to one line of DCL with no temp files created

A good PIPE example

```
$! Counting records in a file
$!!cnt.com
$!!  usage:  @cnt <filename>
$!!
$ if p1 .eqs. "" then inquire/nopunctuation p1 "Enter Filename: "
$ PIPE ( SEARCH 'p1' somegarbage /statistics | -
         SEARCH SYS$PIPE "records searched" | -
         ( READ SYS$PIPE $TMP$ ; DEFINE/JOB/NOLOG $TMP$ &$TMP$ ) ) ; -
         RECORD_COUNT == -
         F$element(2," ",f$edit(F$trnlnm("$TMP$"),"TRIM,COMPRESS")) ; -
         DEASSIGN/JOB $TMP$
$ WRITE SYS$OUTPUT "''F$Fao("!/ Record count is: !AS",RECORD_COUNT)'"
$ EXIT
$!
$!Example:
$!$ @cnt sys$login:login.com
$!
$!   Record count is:   70
```

DEC/Compaq/HP supplied X-windows command symbols

```
$ decw$utils:decw$define_utils.com
```

- Define 19 symbols for programs in decw$utils:

Atobm	bitmap	bmtoa	xdpyinfo
Xev	xlsatoms	xlsfonts	xlswins
Xmag	xmodmap	xprop	xrdb
Xrefresh	xset	xsetroot	xwd
Xwininfo	xwud	xpr	

DEC/Compaq/HP supplied TCP/IP commands

```
$ sys$startup:tcpip$define_commands.com
```

- Define 31 symbols for TCP/IP utilities in sys$system:

```
arp             finger          ifconfig        netstat
ping            ripquery        route           sysconfig
traceroute      dig             named           ndc
nslookup        nsupdate        xfer            ntpdate
ntpdc           ntpq            ntptrace        mosy
snmpi           snmp_request    snmp_traprcv
snmp_trapsnd    uudecode        uuencode        iptunnel
lprsetup        metricview      tcpver  ttcp
```

One line SET DISPLAY command

- If your TCP/IP stack does not to an automatic SET DISPLAY command

```
$ set_display == "set display/create-
        /transport=tcpip-
        /node=""''f$element(1," ",-
        f$getdvi("TT:","TT_ACCPORNAM"))`-
        """_"["_"]"

$ write sys$output f$getdvi("TT:","TT_ACCPORNAM")
Host: alpha7.csxt.csx.com              Port: 3263
$ write sys$output f$getdvi("TT:","TT_ACCPORNAM")
143.42.2.34:1023
$! Note above is bad example!
```

vi like editor

- Vile is a vi clone for VMS
  - http://dickey.his.com/vile/
- From OpenVMS FAQ:
  - vile, vim and elvis are all clones of the vi text editor, and all operate on OpenVMS.
  - Versions of vile are available on the Freeware and at:
    - http://www.clark.net/pub/dickey/vile/vile.html
  - vim: vi improved
    - http://www.polarfox.com/vim/
- Included with the GNV version on the OpenVMS web site

Network Topics

- TCP/IP
  - Multinet / TCPware
  - UCX
  - CMU
- DECnet Topics
  - Access control
  - FAL logging
- Remote Procedures
  - NET_DCL.COM
  - REMOTE_DECW.COM
- Remote Network Alerts
  - OPCOM alerts for DECnet network access
  - OPCOM alerts for FTP network access

Multinet

- Our site uses this for the production VAX systems running v5.5-2H4
- Supports PWIdriver for DECnet Plus DECnet over IP
  - I am not familiar with TCPware but I think it does also

CMU

- Carnegie Mellon University
- Freeware IP for VAX/VMS
- Basic IP stack
  - Telnet
  - FTP
  - Finger
  - Supports DECwindows over CMU
- Does not have a PWI driver for DECnet over IP
- Modification have been made by the user community for OpenVMS/VAX 6.x and 7.X

UCX

- TCP/IP Services for OpenVMS
  - Used on the Alpha VMS systems
- May be using it if you have a NAS license
- I would suggest you get the MADGoat FTP program
  - Supports VMS to VMS file transfers of file type other than ASCII and Binary
  - Latest version of UCX have better VMS to VMS file transfer support
- Also, programs for Finger and News reader are available
  - A Finger program is now included with UCX
  - MadGoat Finger and Penn State Finger
  - ANU News, MXRN and others

Access Control

- Watch for default username and passwords
  - Nonprivileged username and password
- May what to disable default access
  - Default access = incoming and outgoing
  - set default access = none
- Then allow access to individual nodes
  - DEFINE NODE XYZ ACCESS BOTH

FAL Logging

- In the SYLOGIN or the DECnet object login file define FAL$LOG to be “1/disable=8
- This is an unsupported feature
- The 1 outputs to the NETSERVER.LOG or where FAL$OUTPUT is defined file name and file type access information
- The disable=8 disables “Poor Man’s Routing”
  - dir node1::node2::node3::
- FAL$OUTPUT is a logical that can be used to specify the name of the log file to create in place of SYS$OUTPUT
  - `$ DEFINE FAL$OUTPUT FAL.LOG`

Undocumented Command

- In ANALYSIS/SYSTEM is an undocumented command (was before v7.x at least)
- `SDA> SHOW LAN`
- This will show a list of packet types associated with the various protocol stacks on that VMS system

NET_DCL.COM

- Two command procedures in one
  - Interactive
  - Network
- Execute only commands that do not require a terminal
- P1 is the remote node to communicate with

Parts of command procedure (1)

- Interactive part
  - Opens network link to itself on the remote node
    - If file does not exist on the remote node the procedure first copies itself to the remote node, and delete itself after use
  - Sends DCL command to remote node for executation
  - Displays output from remote node
  - Waits for another command to send to the remote node

Parts of command procedure (2)

- Network part
  - Command procedure is executated by the TASK DECnet object
  - Command procedure opens network to be able to receive command from calling node.
  - SYS$OUTPUT is redirected to the network link
  - Command is executated
  - Waits for another command to execute

REMOTE_DECW.COM

- This command procedure builds on the idea of NET_DCL.COM
- Start executation of X-windows application on a remote node and display on local X-windows server
- P1 is remote X-windows client node

Other Optional Parameters

- P2 is the X application to execute
  - Default is to start a DECterm and a File View
- P3 is the screen to display the X application on
  - Default is screen 0
- P4 is network transport to use
  - Default is TCP/IP

DECnet Alert

- Article in December 1992 issue of VAX Professional by John McMahon (Fast Eddie)
- The command procedure issued two OPCOM request message for each network access.
- I modified the file to only issue one OPCOM message

How to use

- Command procedure should be world executable
- Add one line to the `SYLOGIN.COM` file for network access

```
$network:
$ netserver$command:==@sys_system:netinfo
$ exit
```

Message contents

- Local Process Name, PID, & Local Username
- Remote Node name and Username
- DECnet link number
- The DECnet object accessed
  - Number and Name if known
- This information is extracted from the logical SYS$NET

OPCOM Message

```
%%%%%%%%%%  OPCOM  24-JUN-1998 10:46:30.86  %%%%%%%%%%
Message from user FAL$SERVER on SADCSA
FAL_14100028, PID: 00000241     Local:  FAL$SERVER
Remote: SRTC4::O5369P
Link:   65535
Object: 17 (FAL,SUBMIT/REMOTE)

%%%%%%%%%%  OPCOM  24-JUN-1998 10:48:03.47  %%%%%%%%%%
Message from user PHONE$SERVER on SADCSA
PHONE_14100029, PID: 00000242   Local:  PHONE$SERVER
Remote: SRTC4::O5369P
Link:   65535
Object: 29 (PHONE)
```

FTP OPCOM message

- For Multinet when the FTP server starts it executes `MULTINET:FTP_SERVER.COM`
- The system manager can modify this for his additions (special anonymous FTP)
- I have added one line to it

```
$ @sys_system:ftp_server
```

- This command files outputs an OPCOM request on each FTP connect

Message Contents

- Process PID
- Local username used
  - If Anonymous then the password used
- Remote node name and IP address

FTP OPCOM Message

```
%%%%%%%%%%  OPCOM  24-JUN-1998 10:56:24.88  %%%%%%%%%%
Message from user O5369 on SADCSA
Arthur_Cochra$8, PID: 00000243
Local:          O5369
Remote:         srtc4.srs.gov (129.58.66.17)

%%%%%%%%%%  OPCOM  24-JUN-1998 10:57:35.50  %%%%%%%%%%
Message from user ANONYMOUS on SADCSA
<*FTP_O5369P@S>, PID: 00000244
Local:          ANONYMOUS       Password:       O5369P@SRTC4.SRS.GOV
Remote:         srtc4.srs.gov (129.58.66.17)
```

SYSMAN

- STARTUP SET OPTIONS
- STARTUP FILE

STARTUP SET OPTIONS

- If STARTUP_P2 is set to 1, YES, or TRUE then show every DCL line in startup is displayed
- STARTUP SET OPTIONS
  - OPTION STARTUP_P2
  - /verify=full "V"
  - /output=file "D"
  - /checkpointing "C"
  - /verify=partial "P"

STARTUP CHECKPOINTING

- I like this as it displays a line about each phase startup is executing
- Also, displays a line for each command procedure that startup is executing

NOTE

- The SYSMAN STARTUP SET OPTIONS does a:
  - `PARAMETERS USE CURRENT`
  - `PARAMETERS SET STARTUP_P2 "C"`
  - `PARAMETERS WRITE CURRENT`
- It does NOT update MODPARAMS.DAT
  - So, next autogen will wipe out your setting
  - You must manually update MODPARAMS.DAT

STARTUP FILE

- SYSMAN uses the database file with keys to read the entries on startup.
- So it does Batch, Direct, and then Spawn entries.
- Nice if it did the Direct entries last so the Batch and Spawn entries could run in parallel with the Direct entries.

SORTING STARTUP FILE

- sort_vms$layered.srt
- sort_vms$layered.com

sort_vms$layered.srt

```
! SORT_VMS$LAYERED.SRT
! Required by SORT_VMS$LAYERED.COM
/collating_sequence=(sequence=ascii,modification=("S"<"D"))
```

sort_vms$layered.com

```
$! SORT_VMS$LAYERED.COM
$!
$ set noon
$ temp = f$element(0,";",f$parse(p1,f$trnlnm("startup$startup_layered"),,,"SYNTAX_ONLY"))
$ temp_file = f$element(0,";",f$parse(".sorted",temp,,,"SYNTAX_ONLY"))
$ fdl_spec = f$element(0,";",f$parse(".fdl",temp,,,"SYNTAX_ONLY"))
$ specification_file = f$element(0,";",f$parse(".srt",f$environment("PROCEDURE"),,,"SYNTAX_ONLY"))
$!
$ analyze_ /rms /fdl /output='fdl_spec' 'temp'
$ edit_ /fdl /nointeractive /analyze='fdl_spec' 'fdl_spec'
$ create_ /fdl='fdl_spec' 'temp_file'
$ sort_ 'temp' 'temp_file'/overlay -
    /key=(pos:1,siz:12) -  ! Phase, primary indexed key
    /key=(pos:13,siz:1) -  ! Mode, not an indexed key
    /key=(pos:14,siz:79) - ! File, secondary indexed key
    /specification='specification_file'
$ rename /log 'temp_file' 'temp'
$ exit
```

- Note underscores on the four commands. This allows the command to be executed and not any global symbol that may have been defined in SYLOGIN.COM or LOGIN.COM

Startup Database

- Two database files used by SYSMAN
  - STARTUP$STARTUP_VMS
    - Used for the VMS startup
    - Do not modify
  - STARTUP$STARTUP_LAYERED
    - When you add a file to SYSMAN it goes here
- Show which database file currently accessing

```
$ sysman
SYSMAN> startup show database
%SYSMAN-I-DATANAME, STARTUP database is STARTUP$STARTUP_LAYERED
SYSMAN> startup set  database STARTUP$STARTUP_VMS
%SYSMAN-I-NEWCOMPFIL, current component file is now
  STARTUP$STARTUP_VMS
SYSMAN> startup show database
%SYSMAN-I-DATANAME, STARTUP database is STARTUP$STARTUP_VMS
```

Startup Phases

- The phases that VMS uses for startup is
  - `SYS$COMMON:[SYS$STARTUP]VMS$PHASES.DAT`
- Phases are
  - INITIAL
  - DEVICES
  - PRECONFIG
  - CONFIG
  - BASEENVIRON
  - LPBEGIN
  - LPMAIN
  - LPBETA
  - END
- Only last four are usually used for user startup
  - LPMAIN is the default phase

Where the SY*.COM file execute

- INITIAL
- DEVICES
  - SYCONFIG
  - SYLOGICALS
  - SYPAGSWPFILES
- PRECONFIG
- CONFIG
  - SYSECURITY
- BASEENVIRON
- LPBEGIN
  - SYSTARTUP_VMS
- LPMAIN
- LPBETA
- END

Startup File Warning

- When a command procedure is added to the SYSMAN STARTUP by default SYSMAN passes the STARTUP_Pn SYSGEN parameters
- By default STARTUP sets STARTUP_P1 to be “FULL” if the SYSGEN STARTUP_P1 is null.
- This could effect your command procedure if it acts on P1 being something besides null
- If you want a blank P1 parameter given to a specific component file, use the command:
  - `SYSMAN> STARTUP MODIFY FILE component.com/PARAM=P1:""`

Shutdown Minimum Minutes

- SHUTDOWN$MINIMUM_MINUTES
  - Logical used by SYS$SYSTEM:SHUTDOWN
    - `How many minutes until final shutdown [0]:`
- AGEN$SHUTDOWN_TIME
  - Logical used by Autogen for SHUTDOWN and REBOOT phases

Cluster Shutdown

- SYSMAN since v6.x now supports a cluster shutdown from one node via SYSMAN
- However if you do not like the SYSMAN method or what to do a cluster shutdown on a v5.5-2 cluster here is a technique
- Create a SHUTDOWN account
- The account has a login command procedure it runs to do the shutdown

Shutdown Account

```
Username: SHUTDOWN                          Owner:  System Shutdown User
Account:  SYSTEM                            UIC:    [1,5] ([DEC,SHUTDOWN])
CLI:      DCL                               Tables: DCLTABLES
Default:  SYS_COMMON:[SYSMGR]
LGICMD:   SYS_MANAGER:SHUTDOWN
Flags:  DisCtlY DefCLI LockPwd Restricted DisNewMail DisMail DisReconnect
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                     Sat Sun
Primary   000000000011111111112222  Secondary 000000000011111111112222
Day Hours 012345678901234567890123  Day Hours 012345678901234567890123
Network:  -----  No access  ------            -----  No access  ------
Batch:    ##### Full access ######            ##### Full access ######
Local:    ##### Full access ######            ##### Full access ######
Dialup:   -----  No access  ------            -----  No access  ------
Remote:   -----  No access  ------            -----  No access  ------
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:           (none)    Pwdchange:      (pre-expired)
Last Login: 21-JUN-1995 16:31 (interactive), 21-JUN-1995 16:31 (non-interactive)
Maxjobs:         0  Fillm:       200  Bytlm:        65000
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:       200  JTquota:       1024
Prclm:           4  DIOlm:        25  WSdef:         1024
Prio:            4  ASTlm:       200  WSquo:         2048
Queprio:         0  TQElm:        20  WSextent:      4096
CPU:        (none)  Enqlm:       750  Pgflquo:      25000
Authorized Privileges:
  CMKRNL    DETACH    EXQUOTA   LOG_IO    NETMBX    OPER      PHY_IO    SECURITY
  SYSNAM    SYSPRV    TMPMBX    WORLD
Default Privileges:
  CMKRNL    DETACH    EXQUOTA   LOG_IO    NETMBX    OPER      PHY_IO    SECURITY
  SYSNAM    SYSPRV    TMPMBX    WORLD
```

Shutdown Procedure

- Checks to make sure the account is on OPA0:
- Loops for all nodes in the cluster and submits itself to a queue on each node
  - One of the first things shutdown does is stop the queues
- The batch job on each node then runs itself as a detached process
- Finally the detached process runs the SHUTDOWN command procedure passing the answers to the questions as parameters

Shutdown Parameter Note

- Note that the P5 and P6 parameters are in the reverse order for the questions that SHUTDOWN would ask if SHUTDOWN is run interactivity

```
p1 = 0          !"How many minutes until final shutdown [integer]"
p2 = "Cluster Shutdown" ! "Reason for shutdown [string]"
p3 := n         !"Do you want to spin down the disk volumes [yes/no]"
p4 := y         !"Do you want to invoke the site-specific shutdown
                                    procedure"
p5 = "later"!"When will the system be rebooted [string]"
p6 := n         !"Should an automatic system reboot be performed
                                    [yes/no]"
p7 := cluster,save      !"OPTIONS: remove_node, cluster_shutdown,
                        reboot_check, save [string]"
```

Startup Crash Dump Analysis

- A procedure to save a crash dump for later analysis and to generate a summary report on the crash dump

```
$ analyze/crash_dump          sys$system:sysdump.dmp        ! databases:pagefile.sys
  copy              sys_common:[sysmgr.scratch]savedump.dmp ! Save dump file
  set output        sys_common:[sysmgr.scratch]sysdump.lis      !Create listing file
  show crash                                ! Display crash information
  show stack                                ! Show curent stack
  show summary                              ! List all active processes
  show process/pcb/phd/reg    ! Display all current process
  show symbol/all                           ! Display system symbol table
  exit
```

System Information

- DISK_SNOOPY.COM
- MACRO_SNOOPY.MAR
- CLUSTER_SNOOPY.MAR

DISK_SNOOPY.COM

- Display more information than `SHOW DEV D`

```
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
DSA100:                 Mounted              0  PAGE_SWAP        42225     3   1
DSA150:                 Mounted              0  USER3            15531     1   1
DSA200:                 Mounted              0  USER1            67188    58   1
DSA300:                 Mounted              0  ARTHUR_SYS       75648   409   1
$1$DKA100:    (SADCSA)  ShadowSetMember      0  (member of DSA100:)
$1$DKA200:    (SADCSA)  ShadowSetMember      0  (member of DSA200:)
$1$DKA300:    (SADCSA)  ShadowSetMember      0  (member of DSA300:)
$1$DKB100:    (SADCSA)  ShadowSetMember      0  (member of DSA100:)
$1$DKB200:    (SADCSA)  ShadowSetMember      1  (member of DSA200:)
$1$DKB300:    (SADCSA)  ShadowSetMember      0  (member of DSA300:)
$1$DKB400:    (SADCSA)  ShadowSetMember      0  (member of DSA150:)
$1$DKB500:    (SADCSA)  ShadowSetMember      0  (member of DSA150:)
```

Procedure Output

```
Node SADCSA - VAXstation 3100-M76/SPX VMS V7.1     Booted 11-JUN-1998 16:09:36
 Disk Name          Label      Type    Total    Used    %    Free    %  File  Er
--------------  ----------    -----  -------  ------  ---  ------  ---  ---  --
DSA100:         PAGE_SWAP     RZ24    409792  367567  89%   42225  10%    1   0
DSA150:         USER3         RZ24    409792  394261  96%   15531   3%    1   0
DSA200:         USER1         RZ24    409792  342604  83%   67188  16%   56   0
DSA300:         ARTHUR_SYS    RZ24    409792  334144  81%   75648  18%  701   0
$1$RCD0:                      TCP/IP
NFS0:                         TCP/IP
```

Procedure Operation

- Use F$DEVICE to loop information about all disk on a system
- Use F$GETDVI to get the information about a device
- Calculate the blocks used on a disk
- Do integer arithmetic to get the percentages for free and used blocks

MACRO_SNOOPY.MAR

- This program came from a FORTRAN program I am not sure of the origin
- Converted to a command procedure
  - Slow executation
- In the VMS Guide to Performance there is an example of a command procedure to show working set values for processes
  - WORKSET.COM

Conversion to MACRO

- The calls for the working set information was added to the SNOOPY command procedure
- Convert to a MACRO program for speed increase
  - Wanted to program a macro program
  - All VMS systems have a MACRO compiler
- Add code to get processes from all processed in the cluster
- Output a line for each node
  - Macro does 64 bit time arithmetic to get the uptime of the node
  - Could not do this in command procedure

Additions Needed

- Need to add CLD and calls to CLI routines
  - `/NODE[=(nodename,...)]`
  - `/CLUSTER`
  - `/WIDTH=80 or 132`
    - `for 80 columns drop some information`
  - `/OUTPUT=file`
- If you add the additions send me the code

CLUSTER_SNOOPY.MAR

- Macro program to display information on nodes in a cluster
- Quick information instead of the `SHOW CLUSTER` command
- Output is 80 columns
- Macro used because there is no easy way to to the 64-bit time subtraction for the uptime
  - Uptime = Current time - Boot time

Program Output

- Note this is a 80 column output

```
Node         Processor            VMS       Boot Time       Up Time   DECnet Votes
SRTC4    AlphaServer 2100A 5/300 V7.1  12-JUN-1998 06:33  14 08:47    7.263  1
SLCPT1   VAXstation 4000-60       V7.1  12-JUN-1998 06:33  14 08:47    7.300  1
SRTC3    AlphaServer 2100A 5/300 V7.1  12-JUN-1998 06:36  14 08:44    7.262  1
SRTC1    DEC 3000 - M800         V7.1  12-JUN-1998 06:42  14 08:38    7.058  0
SRTC2    DEC 3000 Model 500      V7.1  12-JUN-1998 06:42  14 08:38    7.109  0
SLFLOW   VAXserver 3100          V7.1  12-JUN-1998 06:50  14 08:29    7.169  0
SRTC5    AlphaServer 4100 5/400 4V7.1  24-JUN-1998 16:59   1 22:21    7.307  0
```

Logical Names Outline

- Logical Name Table Outline
  - Normal Logical Name Search List.
  - Adding our Logical Name.
  - Do as Digital says, not as they do.
- Pseudo Disk Outline
  - What is a Pseudo Disk (i.e., Rooted Logical)?
  - What does a Pseudo Disk look like?
  - How can they help the System Manager?
- SYS$SYSROOT Outline
  - How can they help the System Manager?
  - Use with SYS$SYSROOT.

Logical Name Table Outline

- Normal Logical Name Search List.
- Adding our Logical Name.
- Do as Digital says, not as they do.
- Summary

Logical Name Tables

- Two Logical Name Directories
  - Process - LNM$PROCESS_DIRECTORY
  - System - LNM$SYSTEM_DIRECTORY

```
$ show logical/table=lnm$system_directory lnm$directories
   "LNM$DIRECTORIES"  = "LNM$PROCESS_DIRECTORY" (LNM$SYSTEM_DIRECTORY)
        = "LNM$SYSTEM_DIRECTORY"
```

Logical Name Search Table

- Normal Search List is
  - Process
    - LNM$PROCESS LNM$PROCESS_TABLE
  - Job
    - LNM$JOB LNM$PROCESS_hexid
  - Group
    - LNM$GROUP LNM$GROUP_000uic
  - System
    - LNM$SYSTEM LNM$SYSTEM_TABLE

```
$ show logical/full/table=lnm$system_directory lnm$file_dev
   "LNM$FILE_DEV" [super] = "LNM$PROCESS" (LNM$SYSTEM_DIRECTORY)
        = "LNM$JOB"
        = "LNM$GROUP"
        = "LNM$SYSTEM"
1  "LNM$SYSTEM" [kernel,no_alias] = "LNM$SYSTEM_TABLE" [terminal] (LNM$SYSTEM_DIRECTORY)
```

Trusted Logical Name Search Table

- Some VMS Programs only use trusted logicals
- LNM$FILE_DEV has executive mode

```
$ show logical/full/table=lnm$system_directory/access_mode=e lnm$file_dev

   "LNM$FILE_DEV" [exec] = "LNM$SYSTEM"
1  "LNM$SYSTEM" [kernel,no_alias] = "LNM$SYSTEM_TABLE" [terminal] (LNM$SYSTEM_DIRECTORY)
```

Warning! Danger Will Robinson!

- In the next slides about logical names I will talking about modifying logicals setup by VMS. In these cases I only add to search list.
- However, production code on your system may not be certified if changes are made to system logical names.
- Also, TEST, TEST, TEST
- So even if you think changing these logicals is crazy at least learn more about logical names.
- Maybe try these as process logicals for you account only

Add to the Search List

- Want to add our own logical name to search
- Use logical name table LNM_SYSTEM

```
$ show logical/full/table=lnm$system_directory lnm$file_dev
   "LNM$FILE_DEV" [super] = "LNM$PROCESS" (LNM$SYSTEM_DIRECTORY)
        = "LNM$JOB"
        = "LNM$GROUP"
        = "LNM_SYSTEM"
        = "LNM$SYSTEM"
        = "DECW$LOGICAL_NAMES"
1  "LNM_SYSTEM" [exec,table] = "" [terminal] (LNM$SYSTEM_DIRECTORY)
1  "LNM$SYSTEM" [kernel,no_alias] = "LNM$SYSTEM_TABLE" [terminal] (LNM$SYSTEM_DI
RECTORY)
1  "DECW$LOGICAL_NAMES" [exec,table] = "" [terminal] (LNM$SYSTEM_DIRECTORY)
```

Trusted Logical Name Search Table

- Also need to add this table to the executive list

```
$ show logical/full/table=lnm$system_directory/access_mode=e lnm$file_dev

   "LNM$FILE_DEV" [exec] = "LNM_SYSTEM" (LNM$SYSTEM_DIRECTORY)
        = "LNM$SYSTEM"
1  "LNM_SYSTEM" [exec,table] = "" [terminal] (LNM$SYSTEM_DIRECTORY)
1  "LNM$SYSTEM" [kernel,no_alias] = "LNM$SYSTEM_TABLE" [terminal] (LNM$SYSTEM_DI
RECTORY)
```

Logical Name created with DCL

- Use the CREATE/NAME_TABLE command
- Make the system directory the parent
- Make it world readable

```
$ if f$trnlnm ("lnm_system","lnm$system_directory",,,,"TABLE") .eqs. ""
$ then
$  create/name_table/nolog/protection=(system:rwed,owner:rwed,group:r,world:r)-
         /executive_mode/parent_table=lnm$system_directory lnm_system
$ endif
```

- Change to table to lnm$process_directory in the above and define lnm$file_dev in you process or job logical name table to test

Add the to the search list with subroutine

- See handout
- This routine adapted from `SYS$MANAGER:DECW$LOGICALS.COM`
- DECwindows adds its logical name to the search list

Difference between lexicals

- F$TRNLNM
  - Used LNM$FILE_DEV for its logical name search list.
- F$LOGICAL
  - Search Process, Job, Group, and System lists ONLY!

Do as Digital says not as they do

- Digital says in documentation F$LOGICAL is obsolete (if you can find F$LOGICAL)
- Yet, F$LOGICAL is used by several Digital command procedures
- Bit me when I defined DECnet NET* logicals to move the DECnet database files from SYS$SYSTEM.
  - DECnet Phase IV

OpenVMS 7.2 Clusterwide Logical Names

- V7.2 has added cluster wide logical names
  - Cluster member booting gets copy of cluster logical names
- LNM$SYSCLUSTER_TABLE
  - The name of the clusterwide system logical name table.
  - Contains logical names that are available to all users of the cluster.
  - LNM$SYSCLUSTER - logical for LNM$SYSCLUSTER_TABLE
- LNM$CLUSTER_TABLE
  - The parent table for all clusterwide logical name tables
    - including LNM$SYSCLUSTER_TABLE
  - LNM$CLUSTER - logical for LNM$CLUSTER_TABLE
- LNM$SYSTEM is now a search list
  - LNM$SYSTEM_TABLE
  - LNM$SYSCLUSTER - this is the addition

Summary

- Your own system logical name table can help segregate all those system logical names on a system.

Pseudo Disk Outline

- What is a Pseudo Disk (i.e., Rooted Logical)?
- What does a Pseudo Disk look like?
- How can they help the System Manager?
- Description of command procedure.
- Automatic definition at Startup.
- Automatic Startup and Shutdown.
- Other possible uses.
- Summary.

What is a Pseudo Disk?

- Another name for a Rooted Logical.
- OK, what is a rooted logical?
- A syntax that allows directory trees to be referred to as logical devices and top-level directories.
- Reference
  - "Guide to VMS File Applications"
  - Section 6.3.2- 6.3.5

What a Pseudo Disk looks like

```
$ SHOW LOGICAL/FULL  *_DISK
"USER_DISK" [exec] = "DUA4:[USER_DISK.]" [concealed,terminal]
"SYSTEM_DISK" [exec] = "DUA0:[SYSTEM_DISK.]" [concealed,terminal]
"DECNET_DISK" [exec] = "DUA8:[DECNET_DISK.]" [concealed,terminal]
```

Translation Qualifiers

- "USER_DISK" [exec] = "DUA4:[USER_DISK.]" [concealed,terminal]
  - Note that these logicals have CONCEALED and TERMINAL qualifiers.
  - Logicals must be Executive Mode logicals
    - VMS Mail will not work with Supervisor Mode logicals,
    - Images installed with privileges only used ‘trusted’ logicals.
  - The Digital Manual shows a logical for the device and only using the concealed qualifier.
  - I think this works, but because of the command procedure it does not matter.
  - Also , I think under VMS 4.X the physical device was needed.
  - I have always used both qualifiers and this works for me.

How to create a Pseudo Disk

- Create a directory in the [0,0] directory with the last five characters being _DISK
- Run the command procedure on all nodes in the cluster.
  - Example:

```
$ CREATE/DIRECTORY DUA0:[DECUS_DISK]
$ @SYS$MANAGER:PSEUDO_DISKS DUA0 NOSTART
```

What the Procedure does for the System Manager?

- Reduces modifications to system command procedures.
- All rooted directories are automatically handled.
- Users never need to know physical devices.
- Some amount of security.
- Automatic startup of procedures on Pseudo Disks.
- Automatic shutdown of procedures on Pseudo Disks.

Calling the Command Procedure

- Procedure is called once for each disk on the system.
- Parameters:
  - P1 - The disk to be used
    - Logical or Physical name
  - P2 - start, nostart, shutdown, or null

Command Procedure Description

- P1 is translated with F$TRNLNM to see if a logical was passed.
- If translation is not null then the translation is used.
- Else the P1 parameter is used as is.
- Example:
  - DISK7 is a logical for DUA7:
  - If DISK7 is P1 then translated to DUA7:
  - if DUA7: is P1 then used as is

Description Continued 1

- The [0,0] directory is searched in a loop for all *_DISK.DIR files.
- F$PARSE is used to get the device & name of the directory.
- Then the Pseudo Disk logical is defined.

```
$ DIRECTORY DUA7:[0,0]DECUS_DISK.DIR;1
$ DEFINE/SYSTEM/EXECUTIVE_MODE-
        /TRANSLATION=(CONCEALED,TERMINAL)/NOLOG-
        DECUS_DISK DUA7:[DECUS_DISK.]
```

Description Continued 2

- If P2 was null or STARTUP then the Pseudo Disk is searched for [STARTUP]STARTUP.COM and executed if found.
- If P2 was SHUTDOWN then the Pseudo Disk is searched for [STARTUP]SHUTDOWN.COM and executed if found.

```
$ DIRECTORY DECUS_DISK:[STARTUP]
     SHUTDOWN.COM;1     STARTUP.COM;1
```

Description Continued 3

- The STARTUP.COM procedure allows at boot for:
  - Logicals to be defined,
  - Programs started, or
  - Other necessary startup command procedures started.
- The SHUTDOWN.COM procedure allows at shutdown for:
  - Programs stopped or
  - Other orderly functions to happen.
- Again, reduces editing of SYSTARTUP and SYSHUTDWN by the System Manager.

Calling Procedure from SYSTARTUP_V5.COM

- • Prior to VMS 5.4 call command procedure for each disk on system.
  - - Example:
    - `$ @SYS$MANAGER:PSEUDO_DISKS SYS$SYSDEVICE`
    - `$ @SYS$MANAGER:PSEUDO_DISKS DISK1`
    - `$ @SYS$MANAGER:PSEUDO_DISKS USER2`

Using F$DEVICE

- For VMS 5.4 a new lexical F$DEVICE introduced.
- This can be used to find all disks on the system and call Pseudo Disk command procedure for each disk found.
- This reduces the need to edit SYSTARTUP_V5.COM for any new disks that are added to the system.

```
$LOOP:
$    DISK=F$DEVICE("*","DISK")
$    IF DISK .EQS. "" THEN GOTO EXIT_LOOP
$    @SYS$MANAGER:PSEUDO_DISKS 'DISK'
$    GOTO LOOP
$EXIT_LOOP:
```

More Uses

- New mega-gig disk arrives and you wish to move some users to it. When no users have files open do a:

```
$ BACKUP OLD_PHYSICAL_DISK:[PSEUDO_DISK...]*.*.* -
            NEW_PHYSICAL_DISK:[*...]/BY_OWNER=ORIGINAL
$ @SYS$MANAGER:PSEUDO_DISKS OLD_PHYSICAL_DISK SHUTDOWN
$ @SYS$MANAGER:PSEUDO_DISKS NEW_PHYSICAL_DISK STARTUP
```

- Run the Pseudo Disk command procedure on all nodes to redefine the Pseudo Disk logical clusterwide.
- Users will never know they have been moved.
- Watch out for SHUTDOWN and STARTUP command procedures on pseudo disks.

Pseudo Disk in UAF

- Pseudo Disk device names should be used in the user authorization file so the user default device is the Pseudo Disk.

```
$ SHOW DEFAULT
  USER_DISK:[COCHRANE]
```

- Might affect non-image backups because additional level of directories possible.

Pseudo Disk for DECnet

- Also can be used for DECnet accounts.
  - Define a DECNET_DISK Pseudo Disk.
  - Use it for the device for network accounts in UAF.
- Simple way to move the DECnet account file off the system disk.

  Files copied by FAL$SERVER would not fill up the system disk:
  - `DECNET_DISK:[FAL$SERVER]`
  - `DECNET_DISK:[PHONE$SERVER]`
  - `DECNET_DISK:[MAIL$SERVER]`

Other Possible Uses

- P2 parameter of BACKUP.
- Current procedure does not currently have this feature.
- Functions:
  - Stop processed to close open files,
  - Backup databases,
  - Purge Pseudo Disk.
- Could be used to call a `[STARTUP]BACKUP.COM` to do needed functions before a disk backup is done.

Summary

- Ease the editing of the system management files.
- Allows flexible use & movement of files among disks.
- Automatic definition at startup.
- Automatic startup and shutdown.

Some good user logicals – SYS$SCRATCH

- Change SYS$SCRATCH to use a subdirectory off of SYS$LOGIN

```
$define_scratch_space:
$ directory=f$search("sys$login:scratch.dir") .nes. ""
$ if directory
$ then
$ save_privs=f$setprv("sysnam")
$ scratch_space=f$trnlnm("sys$login")+"[.scratch]"-"]["
$ define/job/executive_mode/nolog sys$scratch 'scratch_space'
$ temp=f$setprv(save_privs)
$ endif
```

- The above will work for nonprivileged users also due the way the /executive_mode qualifier works. No error message is given about the user not having the privilege. In that case a supervisor mode logical is created and the system created executive mode logical is not changed.

Another good user logical – SYS_LOGIN

- Create a rooted logical for SYS$LOGIN
- Use to refer to subdirectories under SYS$LOGIN

```
$ login_directory=f$parse(f$trnlnm("sys$login"),,,,-
  "NO_CONCEAL,SYNTAX_ONLY")-"]["-"].;"+".]"
$ define/job/executive_mode/nolog sys_login -
  'login_directory'    /translation=(concealed)

"SYS_LOGIN" [exec] = "ALPHAX$DKA200:[C3723.]"
```

One More – SYS_SCRATCH

- Like last example create a rooted logical for SYS$SCRATCH
- Use to refer to subdirectories off the scratch space

```
$ scratch_directory=f$parse(f$trnlnm("sys$scratch"),,,,-
"NO_CONCEAL,SYNTAX_ONLY")-"]["-"].;"+".]"
$ define/job/executive_mode/nolog sys_scratch -
'scratch_directory` /translation=(concealed)

"SYS_SCRATCH" [exec] = "ALPHAX$DKA200:[C3723.SCRATCH.]"
```

SYS$SYSROOT Outline

- How can they help the System Manager?
- Description of command procedure.
- Use with SYS$SYSROOT.
- Summary.
- Questions?

Warning! Danger Will Robinson!

- In the next slides about logical names I will talking about modifying logicals setup by VMS. In these cases I only add to the search list.
- However, production code on your system may not be certified if changes are made to system logical names.
- Also, TEST, TEST, TEST, and TEST some more!
- So even if you think changing these logicals is crazy at least learn more about logical names.
- Maybe try these as process logicals for your account only

Logical SYS$SYSROOT

```
"SYS$SYSROOT" = "DSA300:[SYS0.]"
              = "SYS$COMMON:"
1  "SYS$COMMON" = "DSA300:[SYS0.SYSCOMMON.]"
```

- SYS$SYSROOT & SYS$COMMON are Executive Mode logicals.
- First translation is a rooted logical definition.
- SYS$COMMON is a rooted logical also.
- Do NOT define SYS$SYSROOT: to be SYS$SPECIFIC: and SYS$COMMON:
  - Problem on boot with F$PARSE.

New Logical

- SYS_COMMON:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE SYS_COMMON
  'F$TRNLNM("SYS$SYSDEVICE")'[VMS_COMMON.]/TRA
  NSLATION_ATTRIBUTES=(CONCEALED,TERMINAL)
$ CREATE/DIRECTORY/OWNER=SYSTEM
  SYS$SYSDEVICE:[VMS_COMMON]
$ SHOW LOGICAL/FULL SYS_COMMON
   "SYS_COMMON" [exec] = "DSA300:[VMS_COMMON.]
    [concealed,terminal] (LNM$SYSTEM_TABLE)
```

New Definition for SYS$SYSROOT

```
"SYS$SYSROOT"      = "DSA300:[SYS0.]"
       = "SYS$COMMON:"
       = "SYS_COMMON:"
1   "SYS$COMMON"    =
 "DSA300:[SYS0.SYSCOMMON.]"
1   "SYS_COMMON"    = "DSA300:[VMS_COMMON.]"
```

- This is done in SYLOGICALS with a subroutine

  (See extract from my SYLOGICALS for details of how this is done.)

Additional Logicals Defined

```
SYS_MANAGER             = SYS_COMMON:[SYSMGR]

SYS_ SYSTEM             = SYS_COMMON:[SYSEXE]

SYS_ LIBRARY            = SYS_COMMON:[SYSLIB]

SYS_ HELP               = SYS_COMMON:[SYSHLP]

SYS_ EXAMPLES           = SYS_COMMON:[SYSHLP.EXAMPLES]

SYS_STARTUP             = SYS_COMMON:[SYS$STARTUP]
```

- Not required for this technique to work but nice for easy access to files

SYS_MANAGER Usage

- SYS_MANAGER is a place to keep all the various system management command file and programs what the system manager develops.
- This directory will allow the system manager developed files to be kept separate from the Digital and layered product supplied.
- Example would be:
  - `SYCONFIG.COM`
  - `SYLOGICALS.COM`
  - `SYPAGSWPFILES.COM`
  - `SYSECURITY.COM`
  - `SYSTARTUP_V5.COM`
  - `LAT$SYSTARTUP.COM`
  - `SYLOGIN.COM`

Problem

- When system is booting SYS_COMMON is not a part of SYS$SYSROOT yet.
- So, if the startup procedures called by STARTUP.COM are not in SYS$COMMON:[SYSMGR] then they will not be executed.
- How can you move the procedures to SYS_MANAGER: and yet get them executed at boot time?

SY*.COM Files

```
$if f$trnlnm("sys_manager") .eqs. "" then -
   define/process/nolog sys_manager sys$sysdevice:[vms_common]
$name=f$element(0,";",f$parse("sys_manager:",-
                        f$environment("PROCEDURE"),,,"SYNTAX_ONLY"))
$@'name' "'p1'" "'p2'" "'p3'" "'p4'" "'p5'" "'p6'" "'p7'" "'p8'"
$if f$trnlnm("sys_manager","lnm$process") .nes. "" then -
                        deassign/process sys_manager
$exit
```

- Used in SYS$COMMON:[SYSMGR] to ensure our moved files are executed at boot time.
- Note this is one place I had to hard code

SYS_SYSTEM Usage

- SYS_SYSTEM is a good place to keep executable for non-Digital programs. Example would be KERMIT.EXE
- Then the foreign command is easy to use:

```
$ symbol :== $program
```

SYS_LIBRARY Usage

- SYS_LIBRARY is the place to put user developed printer device control text libraries.
- The printer device control text libraries MUST be in SYS$LIBRARY, but since SYS$SYSROOT has been redefined they will be found in SYS_LIBRARY.
- Also, user developed sharable images, since SYS$SHARE=SYS$LIBRARY

SYS_HELP Usage

- SYS_HELP is the place to put user developed help libraries and also application help libraries.
- An example is the help file for KERMIT.

SYS_EXAMPLES Usage

- SYS_EXAMPLES is a good place to but any program source code for the programs in SYS_SYSTEM.

SYSMAN Startup Will Work

```
"SYS$STARTUP"           =
   "SYS$SYSROOT:[SYS$STARTUP]"
                        = "SYS$MANAGER"
1  "SYS$MANAGER"        =
   "SYS$SYSROOT:[SYSMGR]"
```

- Since SYS$SYSROOT has been redefined SYSMAN will find the startup files in the SYS_MANAGER and SYS_STARTUP directories.
- This directory can be used for the user developed procedures which are executed by the SYSMAN startup procedure.
- Starting printer queues, installing images, etc.

Additional Logical for Databases

```
DATABASES =
SYSTEM_DISK:['F$GETSYI("NODENAME")'_ DATABASES ],
SYSTEM_DISK:[COMMON_ DATABASES ]
```

- Node Specific - Specific Network Files & Page/Swap Files.
- Common Files - Queue File, License, Common Network Files, Authorization Files, VMS Mail File, SYSMAN Layered File.
- Logicals can be used to reference these moved files:
  - Sysuaf, Rightslist, Netproxy, VMSmail_profile
  - Netnode_local, Netnode_remote, Netobject
- Also the queue files can be placed here.
  - `QMAN$MASTER.DAT`
  - `SYS$QUEUE_MANAGER.QMAN$JOURNAL`
  - `SYS$QUEUE_MANAGER.QMAN$QUEUES`

Read-only System Disk?

- OR, Files that can not be moved from the system disk.
- If the SYS_COMMON directory were on another disk the the system disk could be almost read only.
- Except for the following files:
  - SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT
  - SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT
  - The files AUTOGEN generates.

Summary

- Ease the editing of the system management files.
- Allows flexible use & movement of files among disks.
- Automatic definition at startup.
- Automatic startup and shutdown.
- Redefining SYS$SYSROOT allows the manager to keep better track of the files on the system.

StorageWorks Command Console

- Provides MS/Win GUI for management of StorageWorks storage array controllers.
  - HSJ (CI)
  - HSZ (SCSI)
  - HSG (FC-SF)
- Uses TCP/IP to communicate with server agent on OpenVMS.
- Behaves like other “Explorer” software.

StorageWorks Command Console

- Limitations:
- PC's IP address must back-translate
  - DHCP is o.k. so long as DNS is updated when address lease is obtained / renewed.
- Does not work over WAN unless PC's DNS name is "visible" outside of firewall and firewall allows the TCP ports.
- OpenVMS server agent will only run on one node of a cluster.

StorageWorks Command Console

- Limitations, cont'd:
- Unit names and storage-set names are assigned randomly and arbitrarily.
  - Some names can be changed manually using the CLI.
- Can hold onto the virtual console so that other access means are denied:
  - SET HOST/DUP, SET HOST/SCSI

StorageWorks Command Console

- Limitations, cont'd:
- Disks falling into the Failed Set are detected and reported as warnings; however, CLI messages are not passed through to the GUI - you must still connect to the CLI to get them.
  - "Other controller restarted"
  - Cache battery alerts

StorageWorks Command Console

- Limitations, cont'd:

- No provisions for running HSx utilities and diagnostics.

- No performance data available via the GUI - use the CLI to run VTDPY.

StorageWorks Command Console

- Management Considerations
- PCs must be authorized to access OpenVMS server agent. Use the SWCC configuration utility supplied with the OpenVMS-side software.

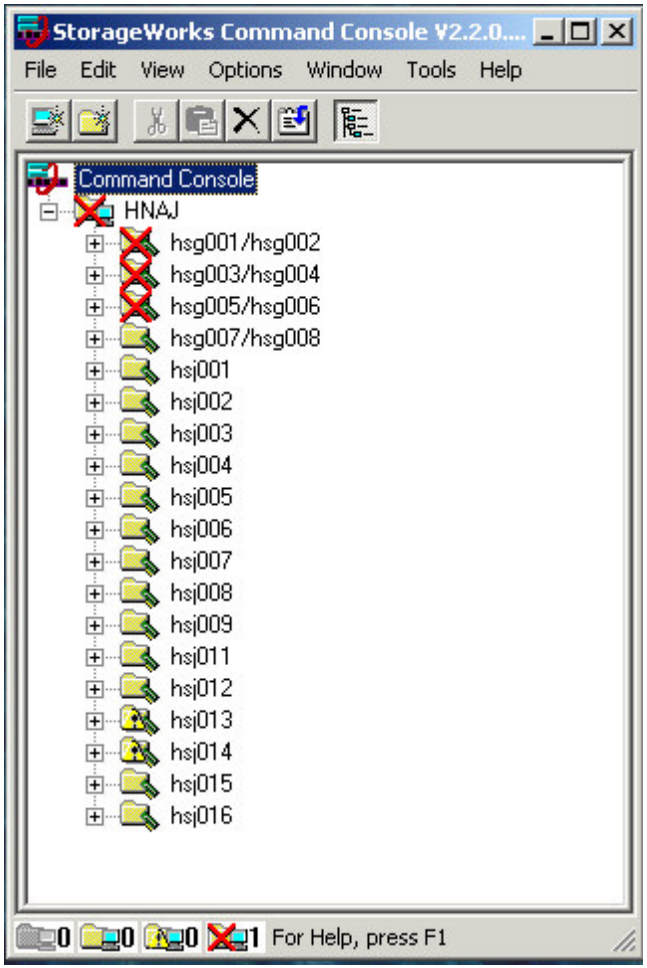
```
$ @sys$manager:swcc$config.com
sys$specific:[swcc$agent]client.ini
```

- Controllers and/or controller pairs must be set up using the SWCC configuration utility supplied with the OpenVMS-side software.

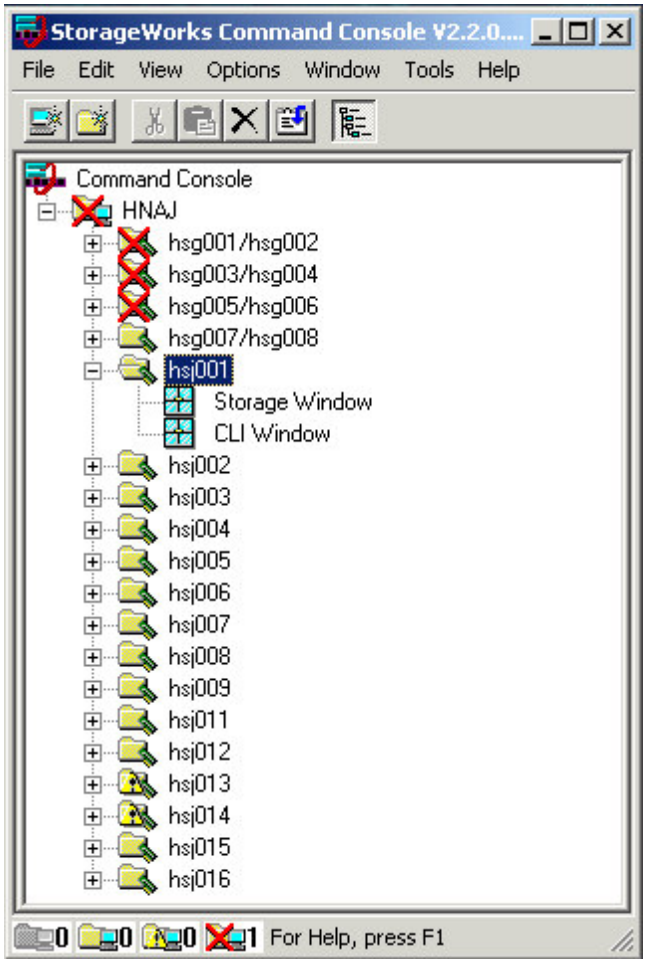
StorageWorks Command Console

- Management Considerations
- HSZ and HSG controller pairs present only a single virtual device for remote access - cannot connect to an individual controller by name using the CLI window.
- You will still need to access the physical console terminal port from time to time, as when a controller fails out of the pair.

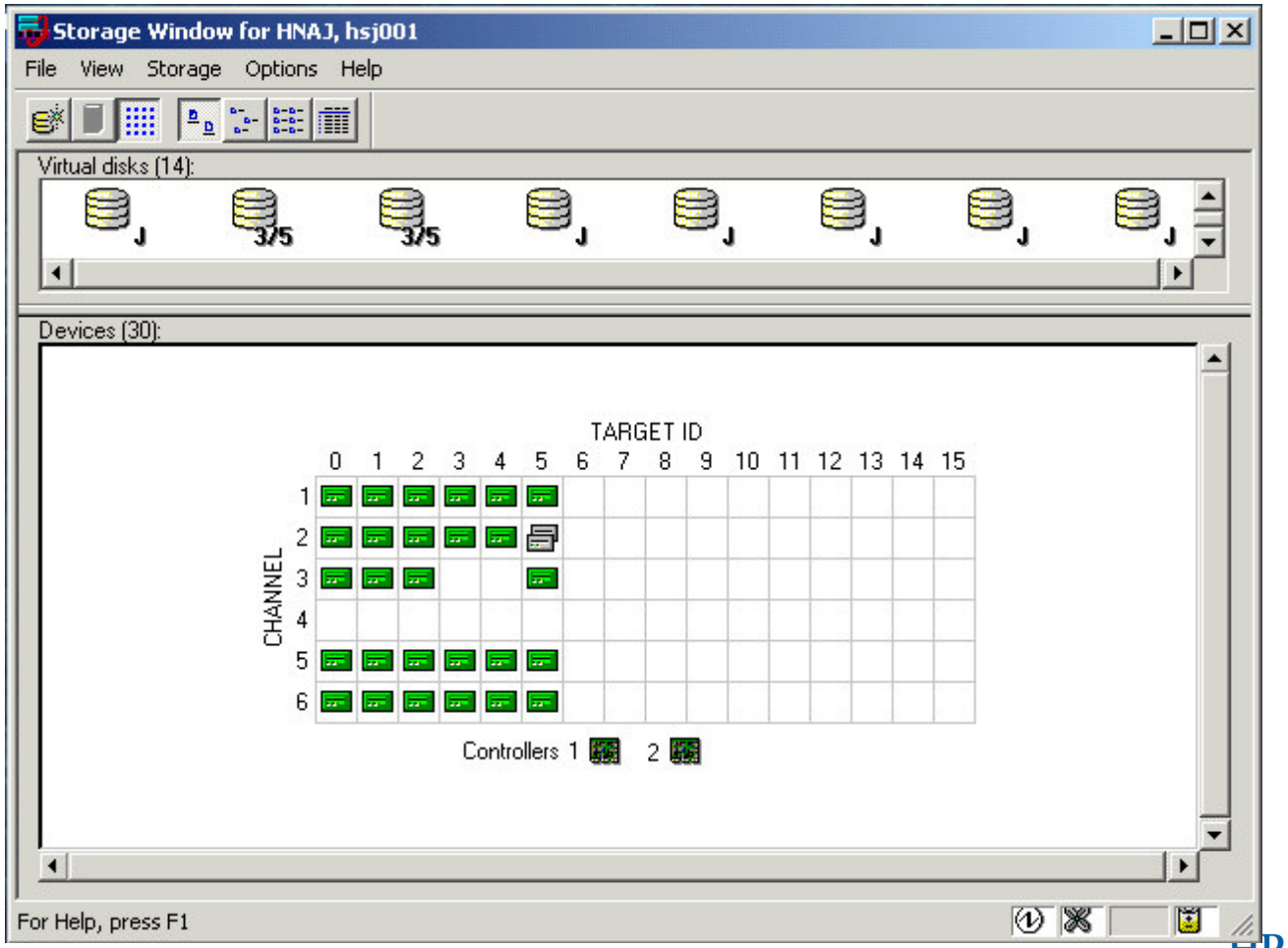
StorageWorks Command Console



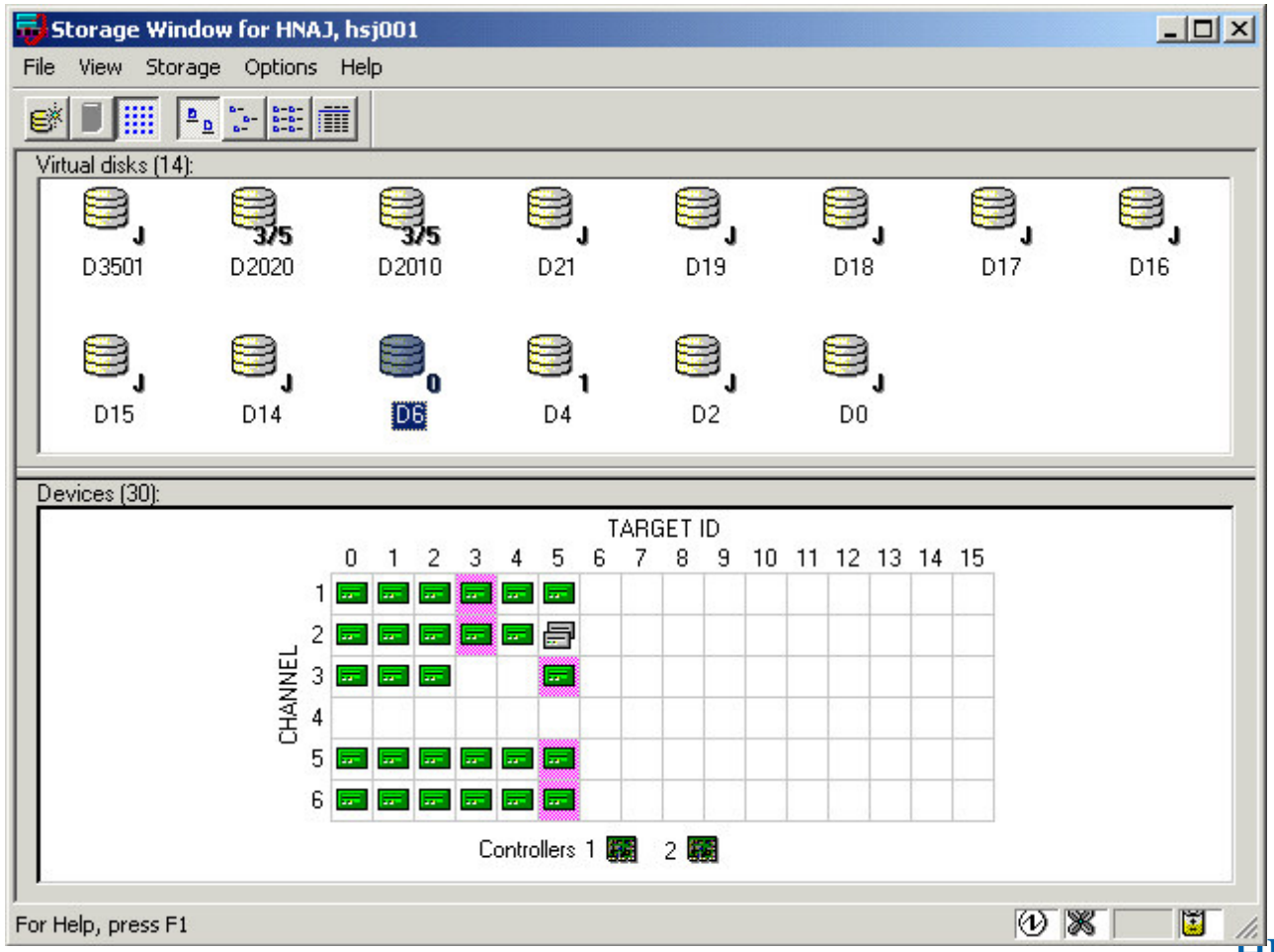
StorageWorks Command Console



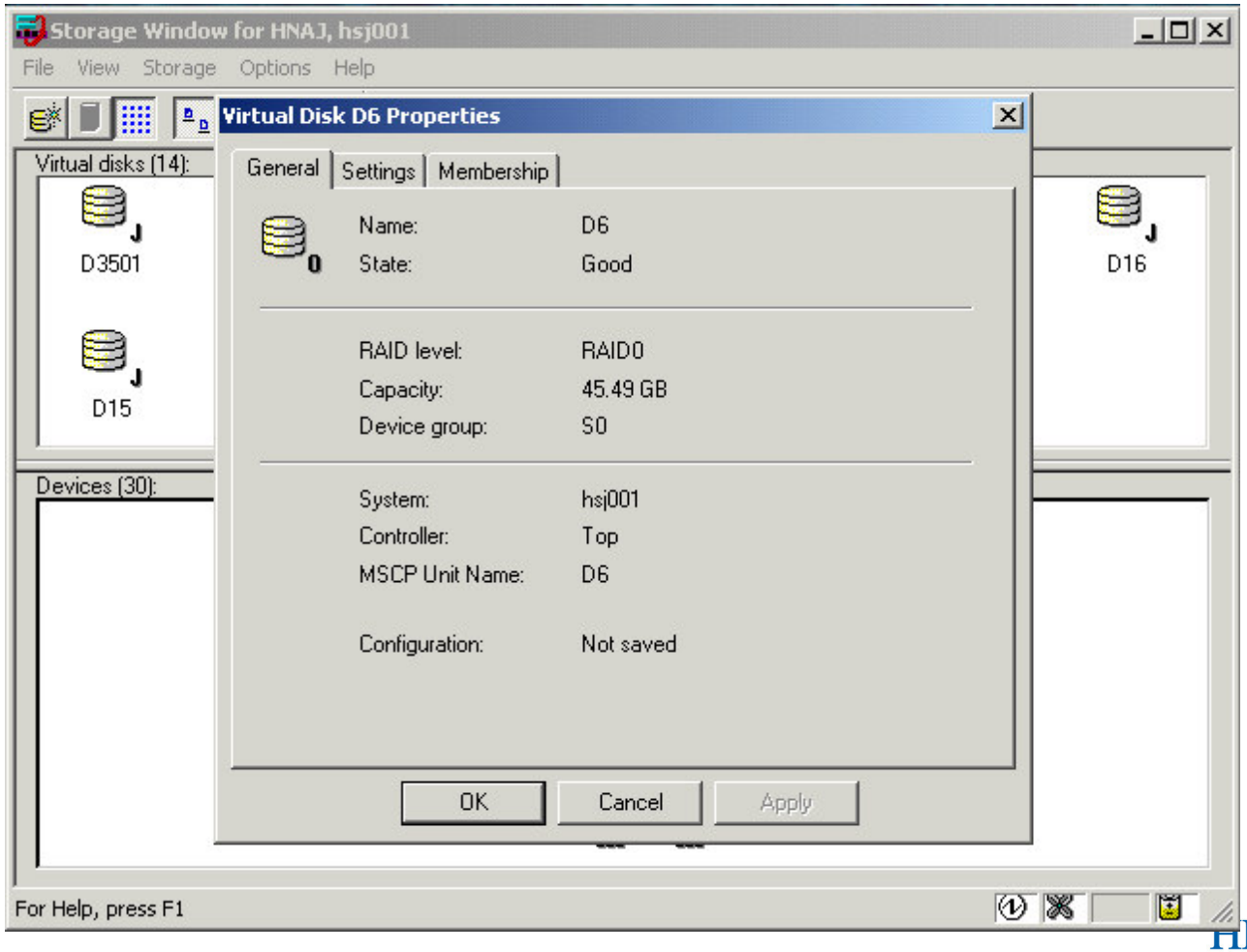
StorageWorks Command Console



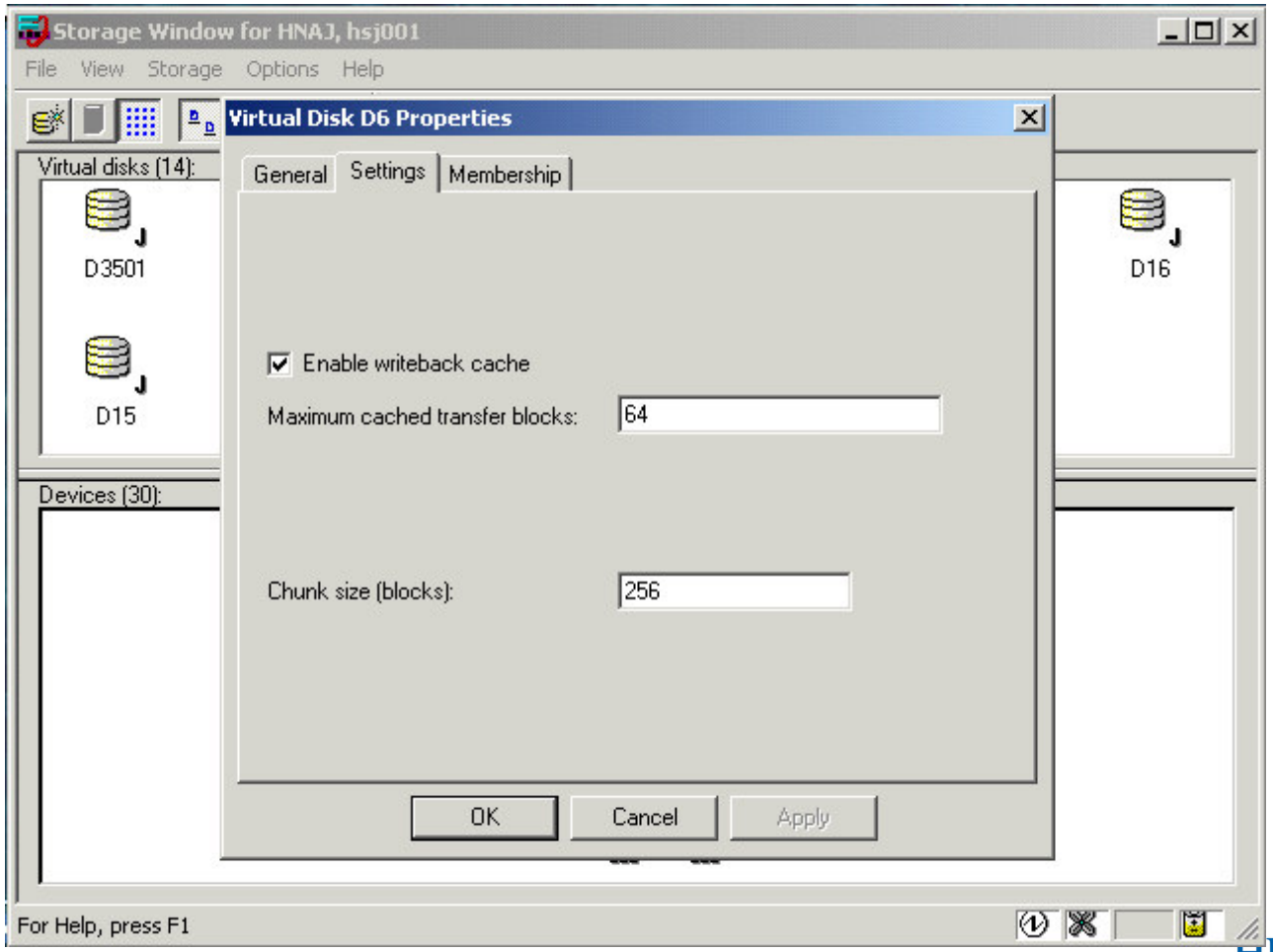
StorageWorks Command Console



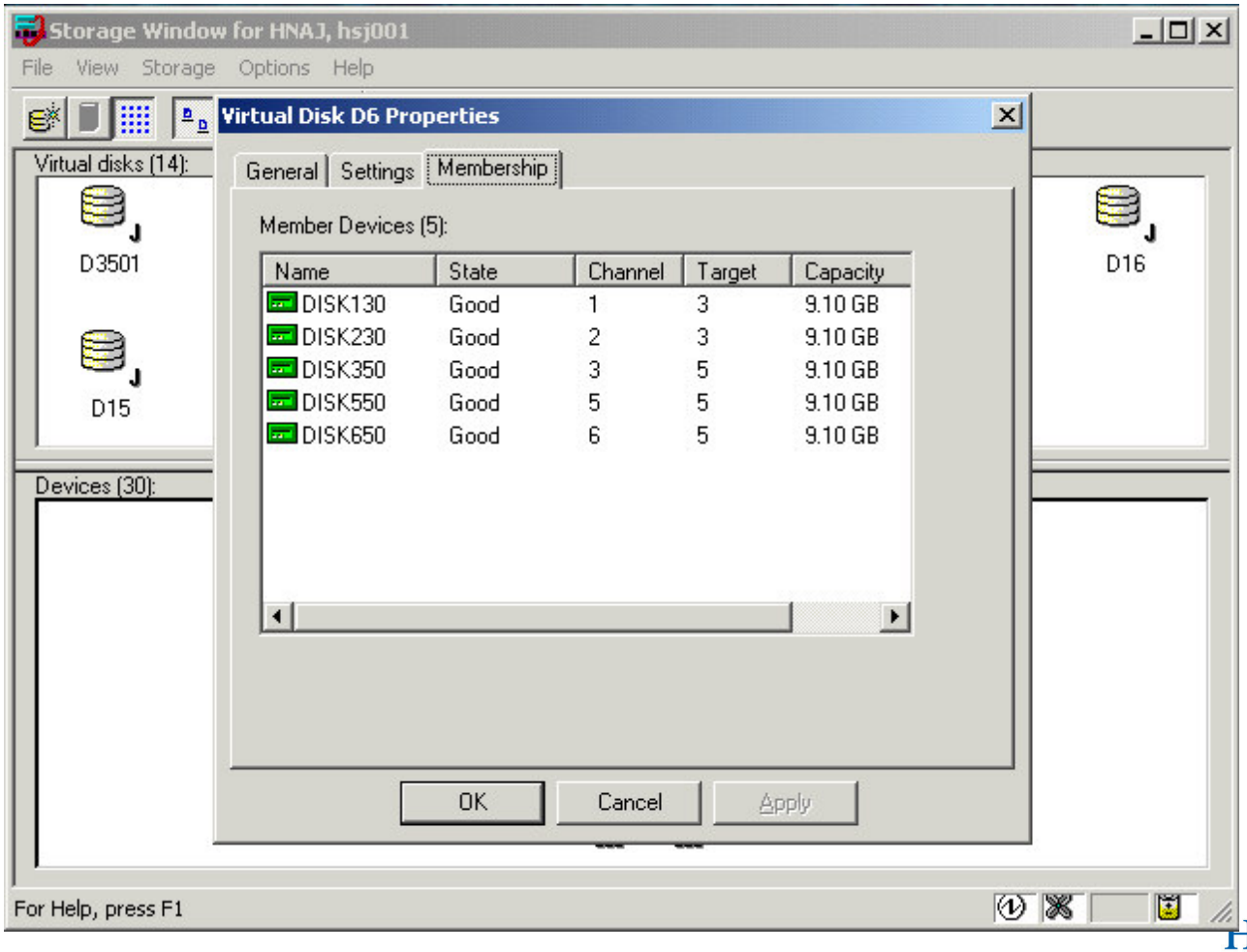
StorageWorks Command Console



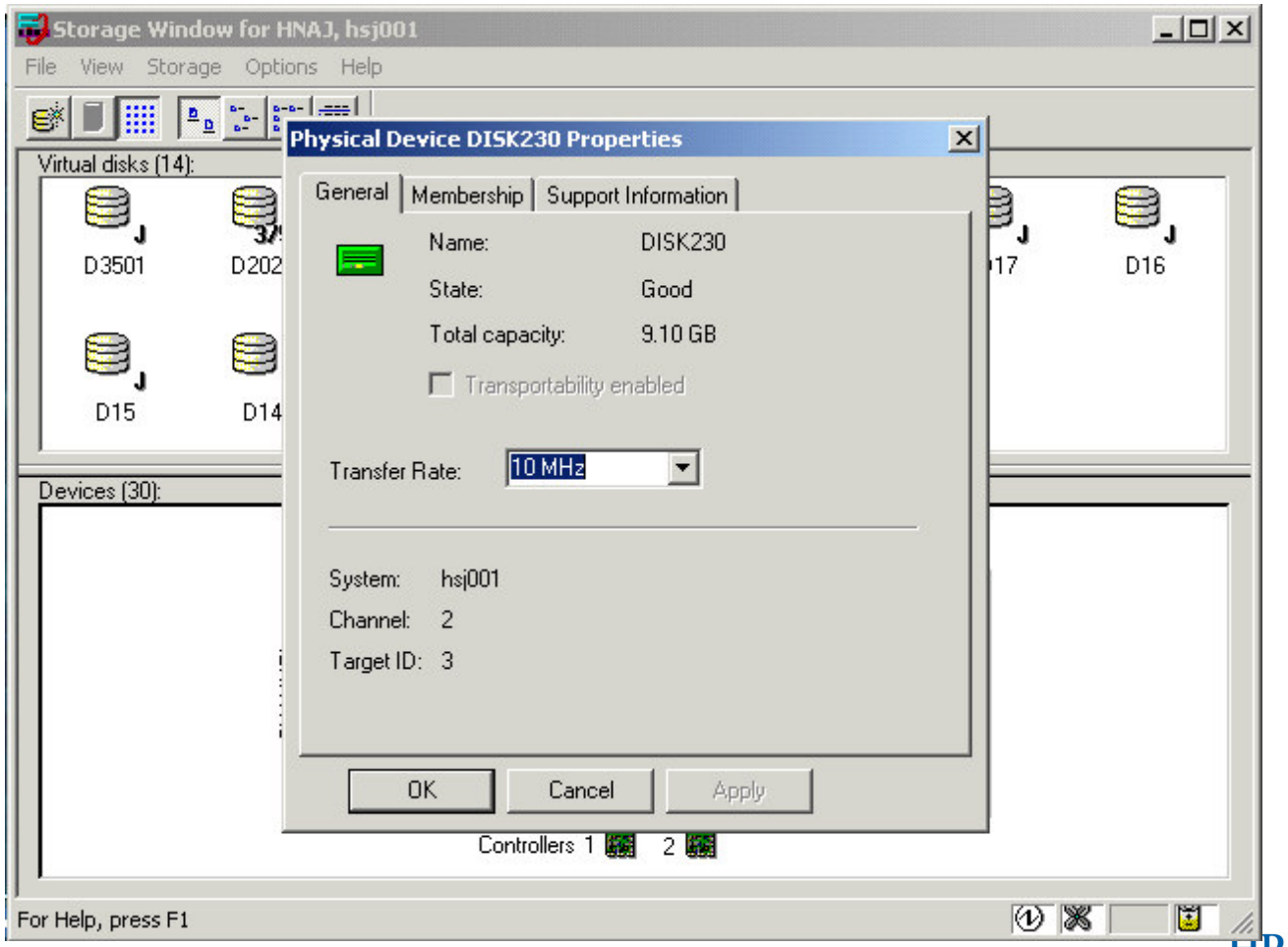
StorageWorks Command Console



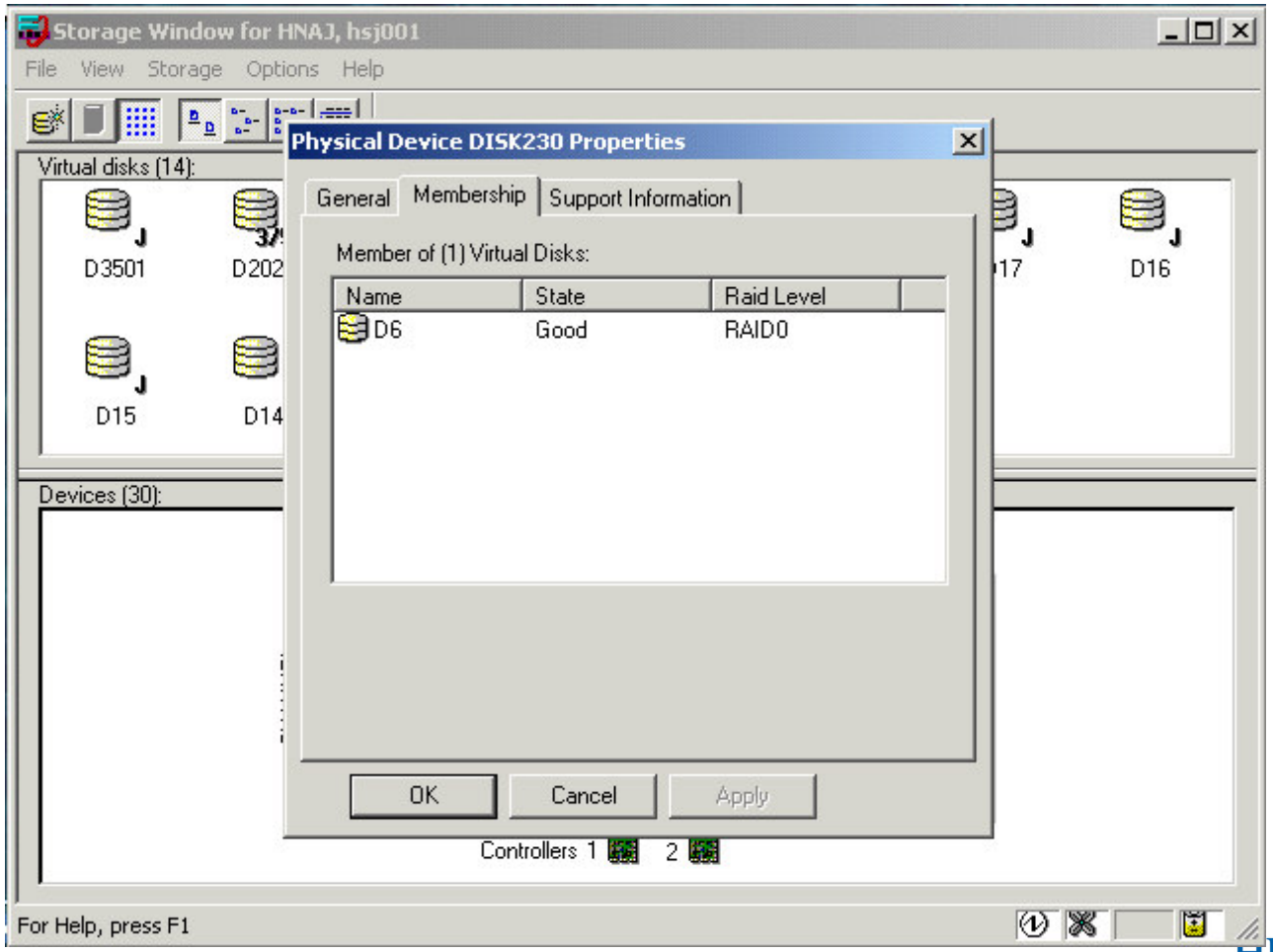
StorageWorks Command Console



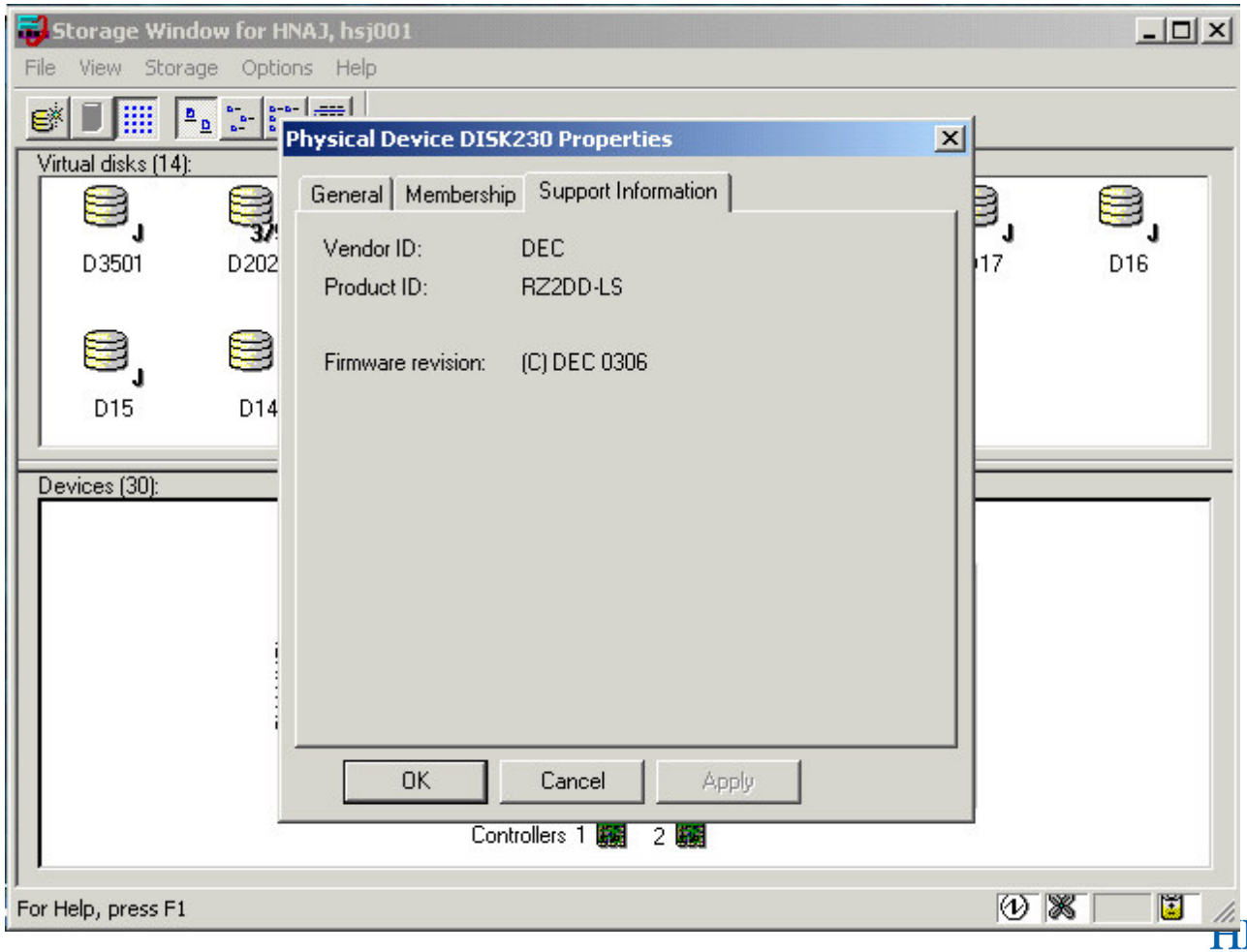
StorageWorks Command Console



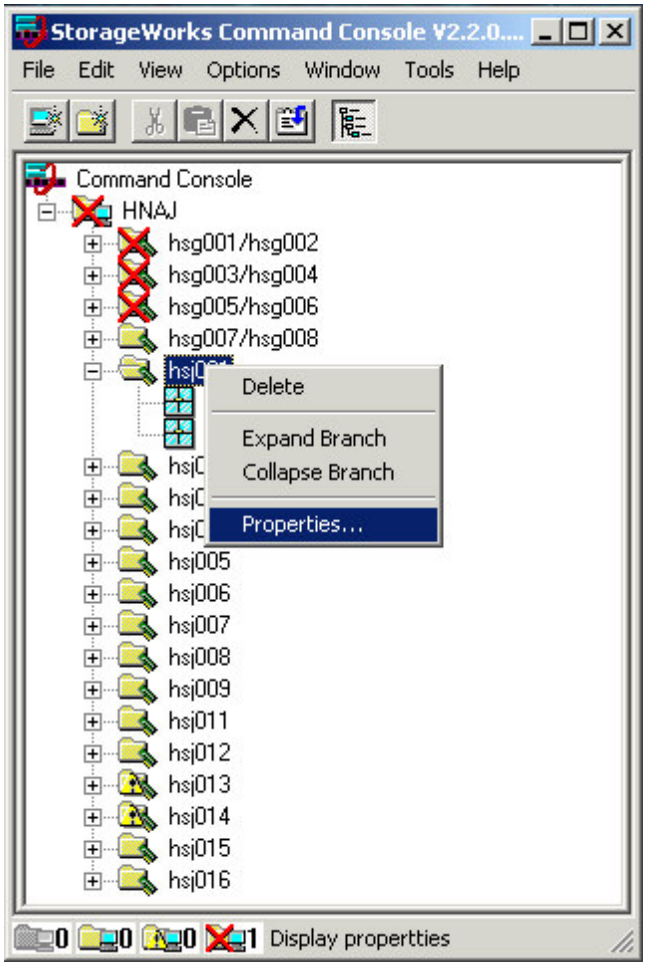
StorageWorks Command Console



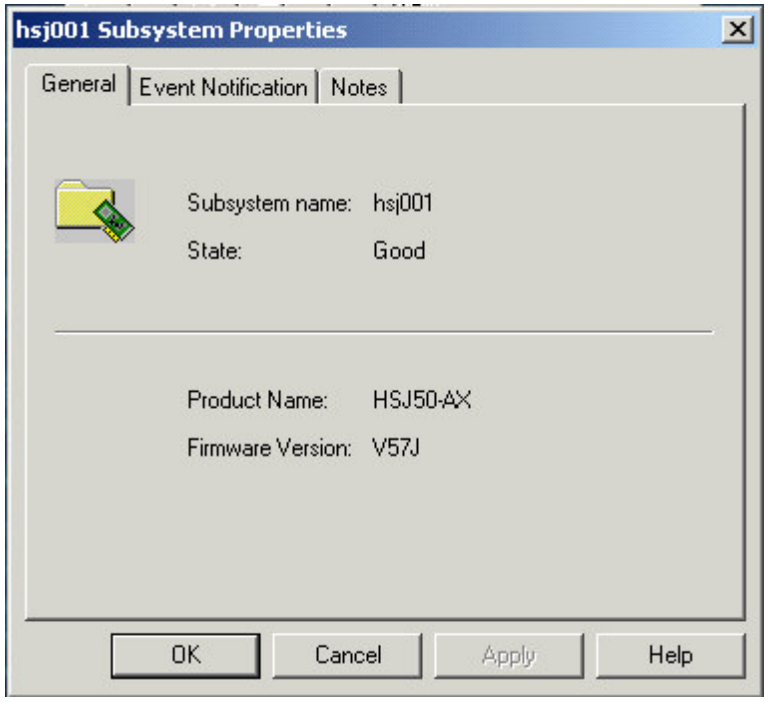
StorageWorks Command Console



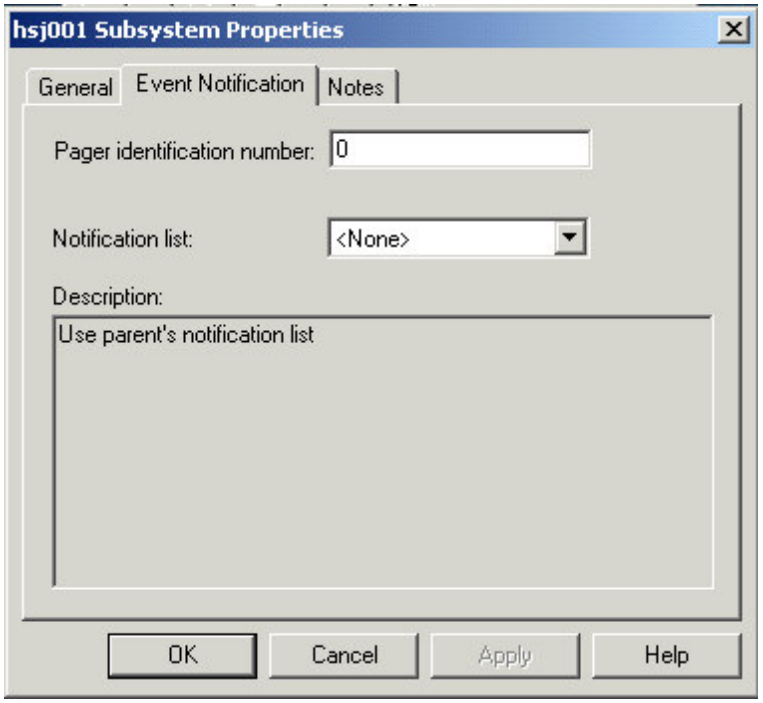
StorageWorks Command Console



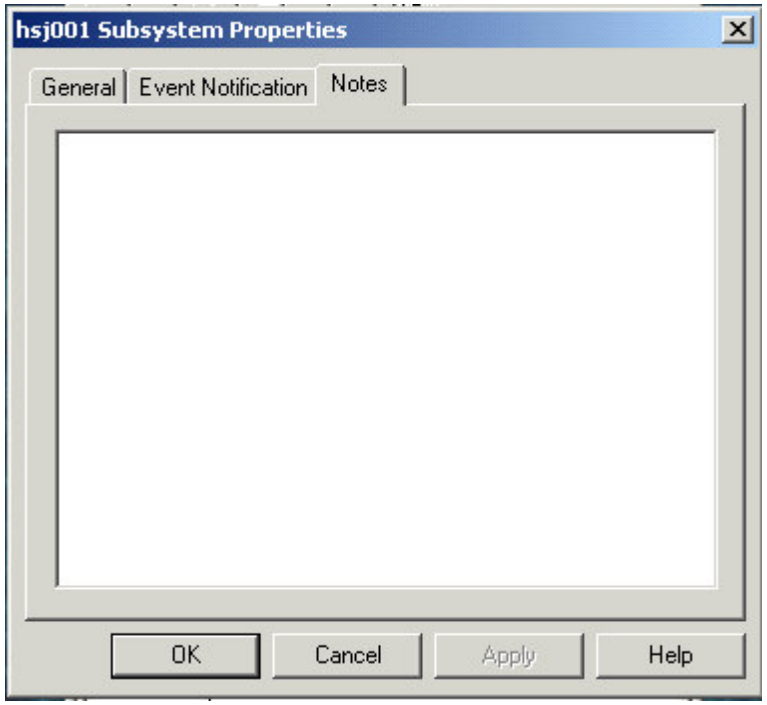
StorageWorks Command Console



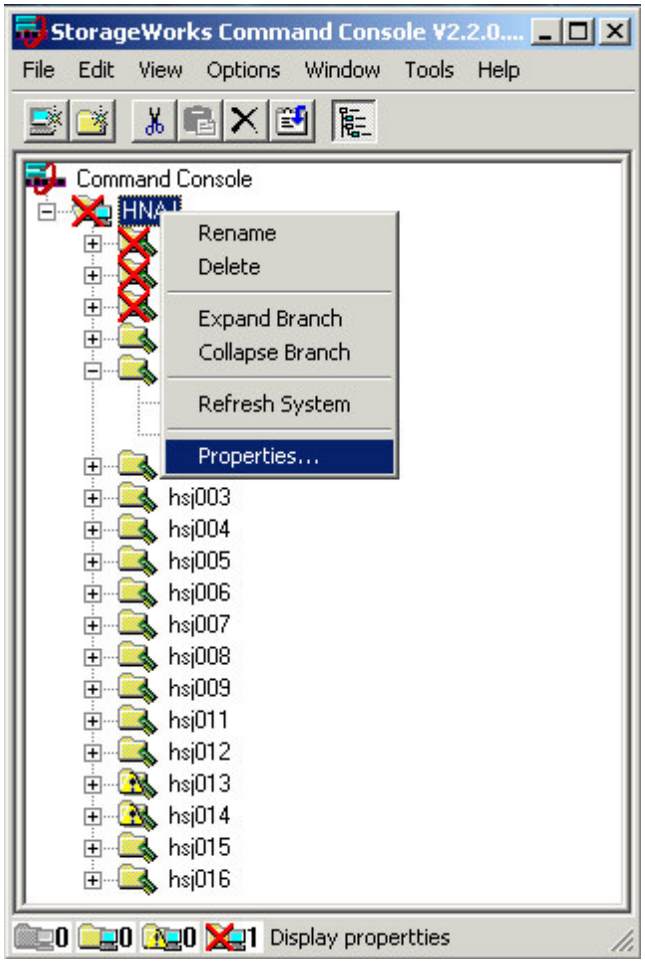
StorageWorks Command Console



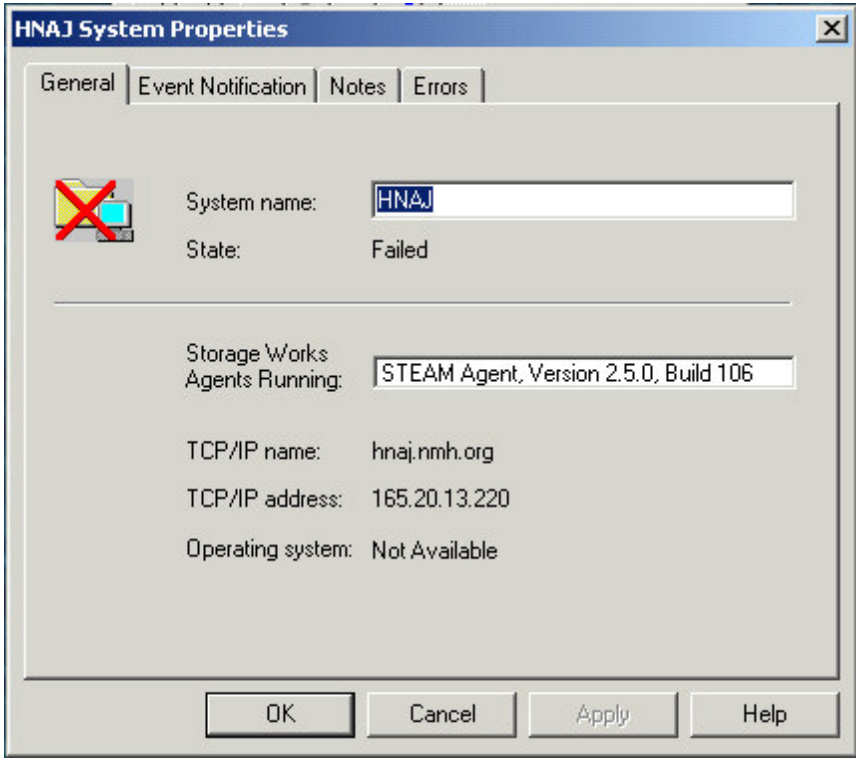
StorageWorks Command Console



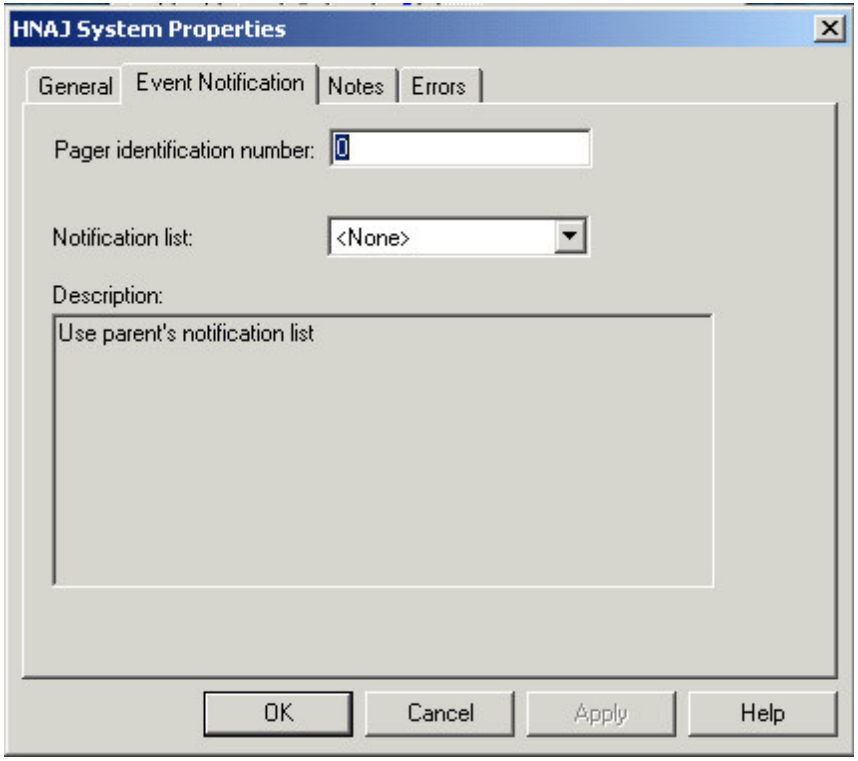
StorageWorks Command Console



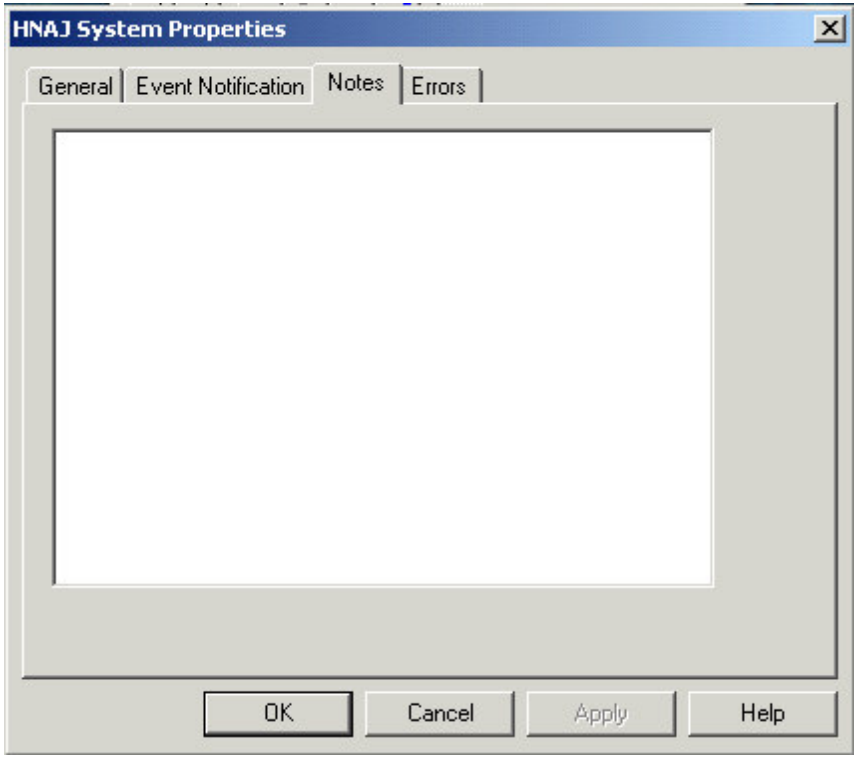
StorageWorks Command Console



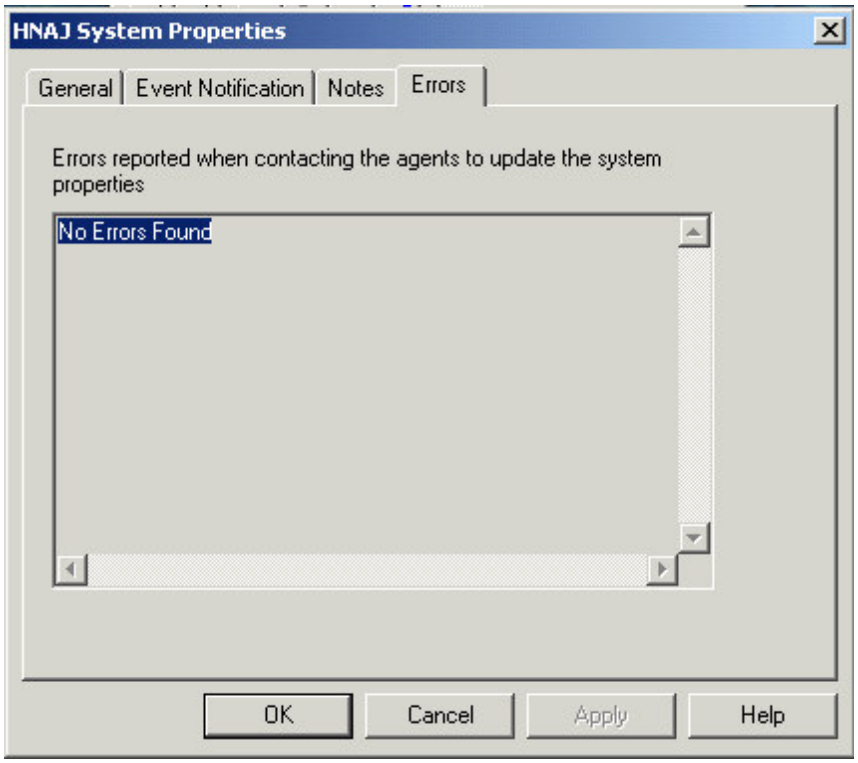
StorageWorks Command Console



StorageWorks Command Console



StorageWorks Command Console



OpenVMS Management Tools

- Digital's Availability Manager for Distributed Systems (DECamds) is a real-time monitoring, diagnostic, and correction tool used by system managers to improve availability. DECamds collects and analyzes data from multiple nodes simultaneously, directing all output to a centralized DECwindows display. The analysis detects resource availability problems and suggests corrective actions.
- DECamds is a Motif based program

Sources for Files

- AMDS and Availability Manager are on the OpenVMS distribution CDs
- Latest versions and documentation on the OpenVMS web site

AMDS Installation

- OpenVMS 6.2 version of AMDS
  - Requires VMScluster license
  - Can be installed on VAX/VMS 5.5-2 also
- OpenVMS 7.1 version of AMDS
  - Only requires a OpenVMS license
  - Can be installed on OpenVMS 6.2 also
- Start the server with
  - `@sys$startup:amds$startup start`
- Also, note AMDS is not routable

AMDS Startup

- Start AMDS with
  - `@sys$startup:amds$startup start`
- Logicals are defined in `AMDS$SYSTEM:AMDS$LOGICALS.COM`
  - AMDS$GROUP_NAME is the group to display the node information in, default is DECAMDS
    - Define a group name for each cluster
  - AMDS$DEVICE is used to define the network device to use if you have multiple LAN connections

AMDS is x-windows based

- AMDS is x-windows Motif based so it needs a x-windows server on the PC client.
- Since Digital/Compaq wants you to use this tool they provide a x-windows server for Windows 9x/NT
- The x-windows server provided is eXcursion
  - Or it was before Availability Manager
  - Check the Pathworks CD that ships with CONDIST

Availability Manager

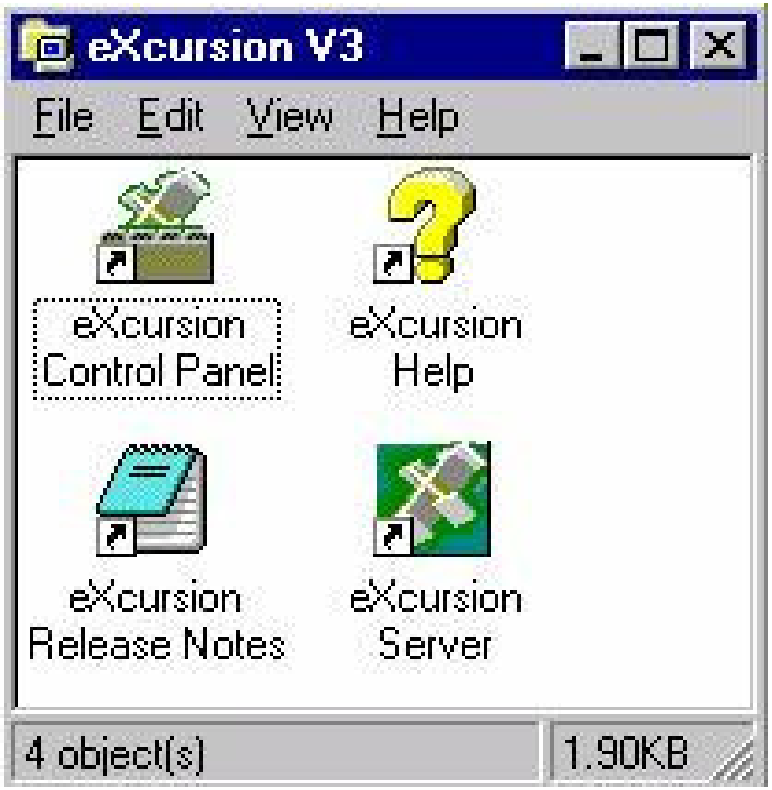
- Windows NT based tool
  - http://www.openvms.compaq.com/openvms/products/availman/download.html
- AMDS is Motif based
  - eXcursion no longer part of PC kit
  - BUT, check Pathworks CD included in distributions
- AMDS server driver now part of OpenVMS 7.2 installation
  - Only install client for Motif based monitoring

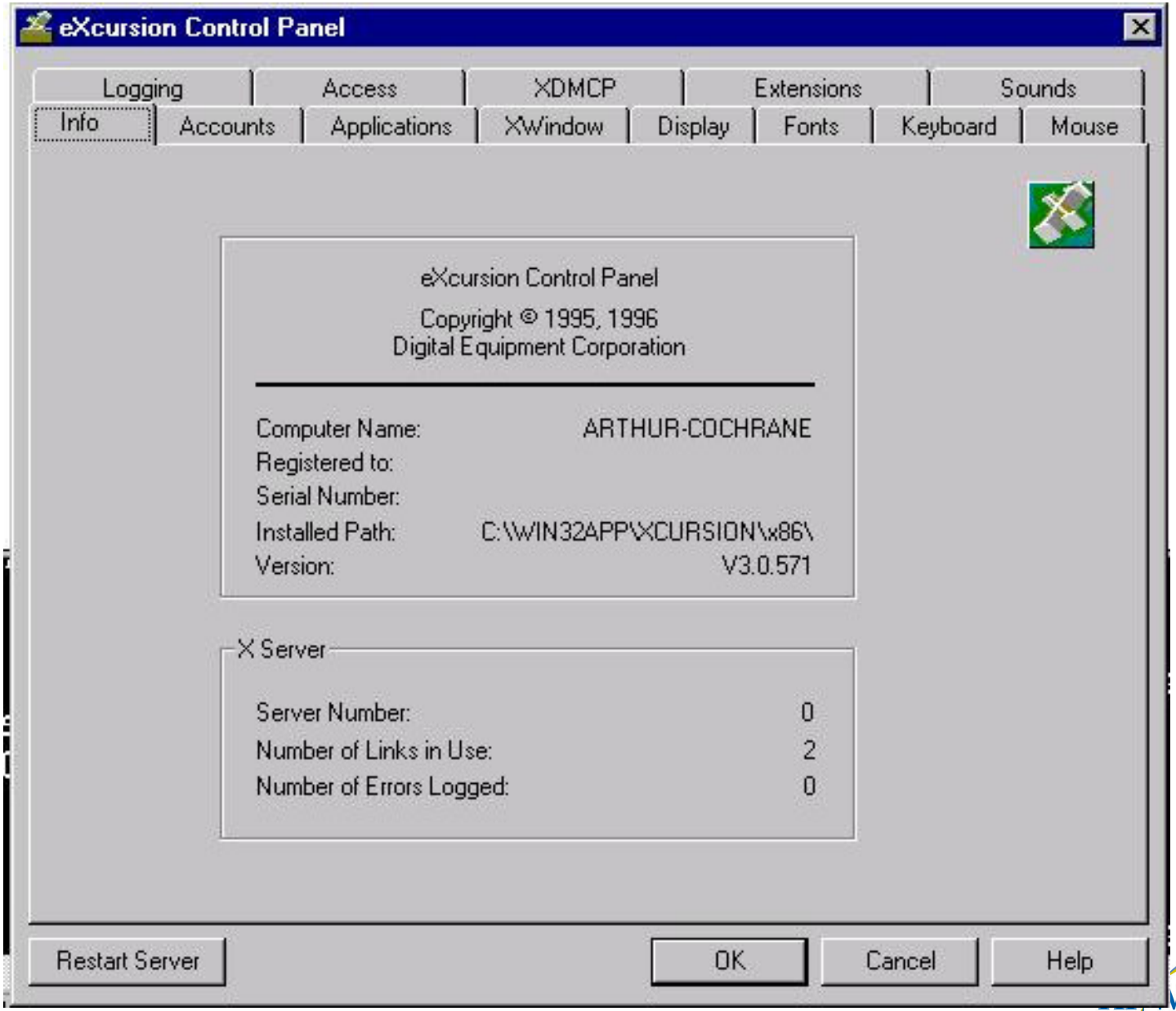
Advantages of AMDS over Availability Manager

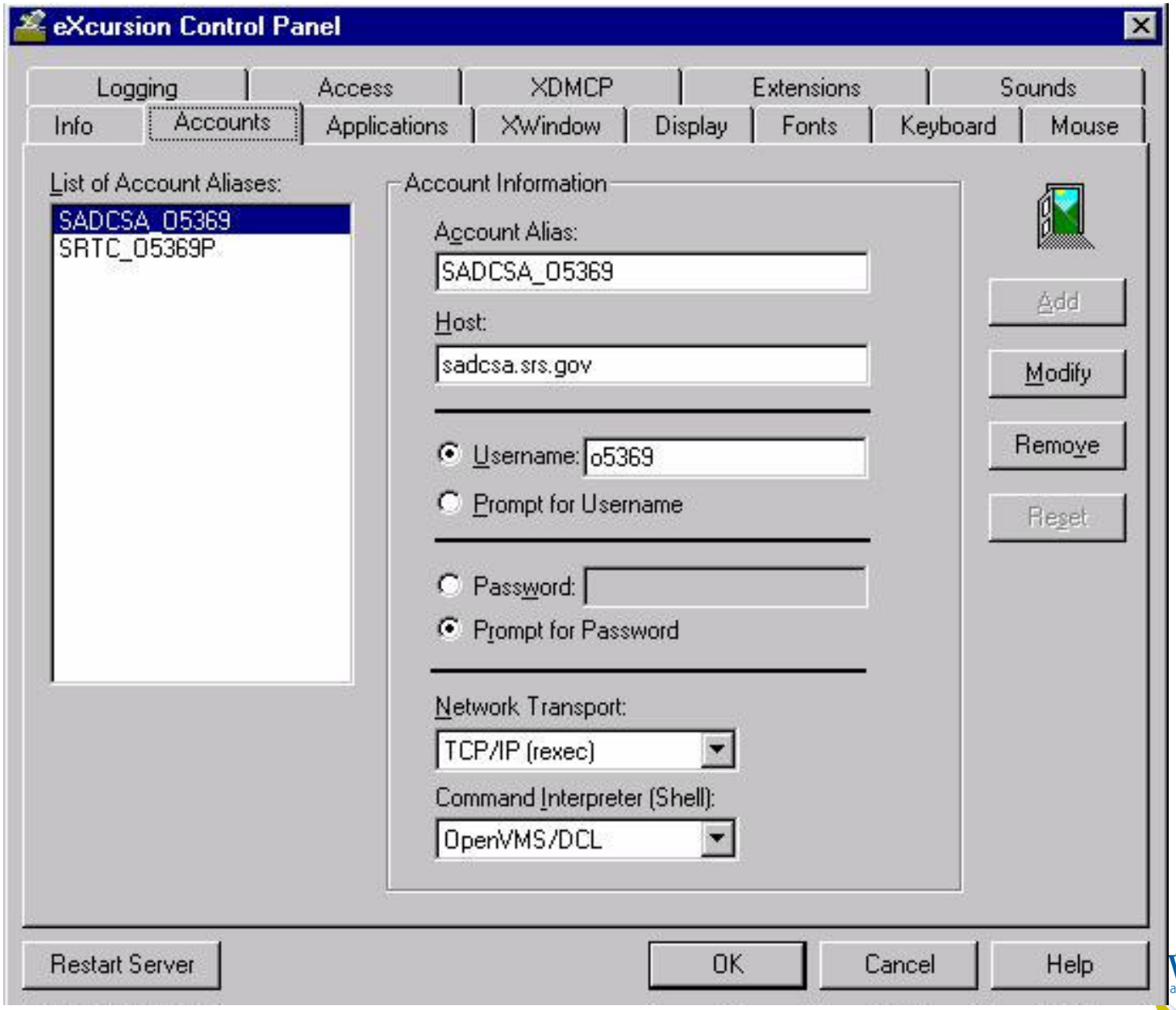
- The protocol is non routable
- The PC running Availability Manager has to be in the same subnet as the VMS systems monitored
- AMDS is X-windows based
- On one of the low use VMS nodes run the collector and with a SET DISPLAY direct the output over DECnet or TCP/IP to a VMS workstation with DECwindows or even a Linux system or PC with eXcursion or Exceed

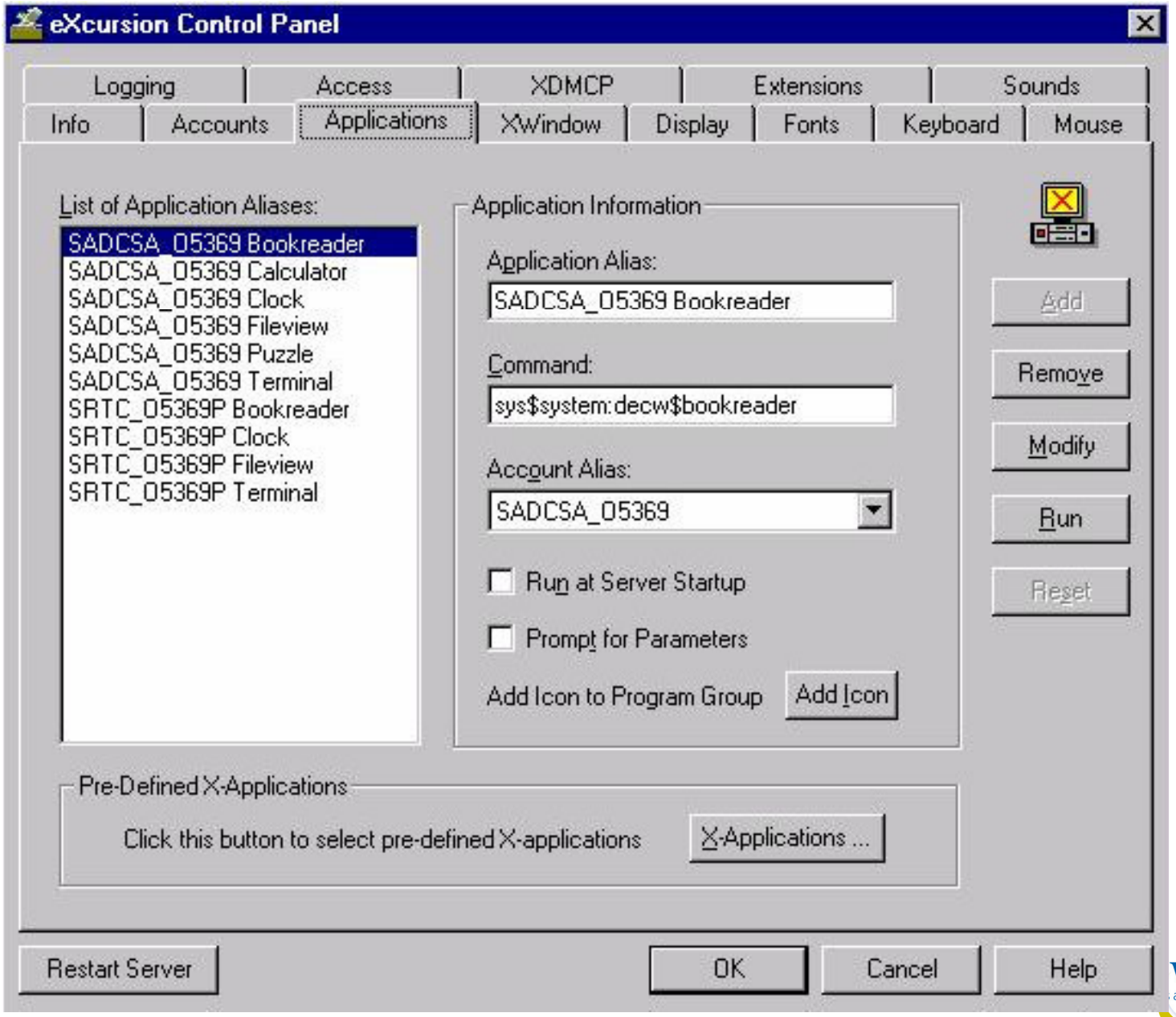
eXcursion Installation

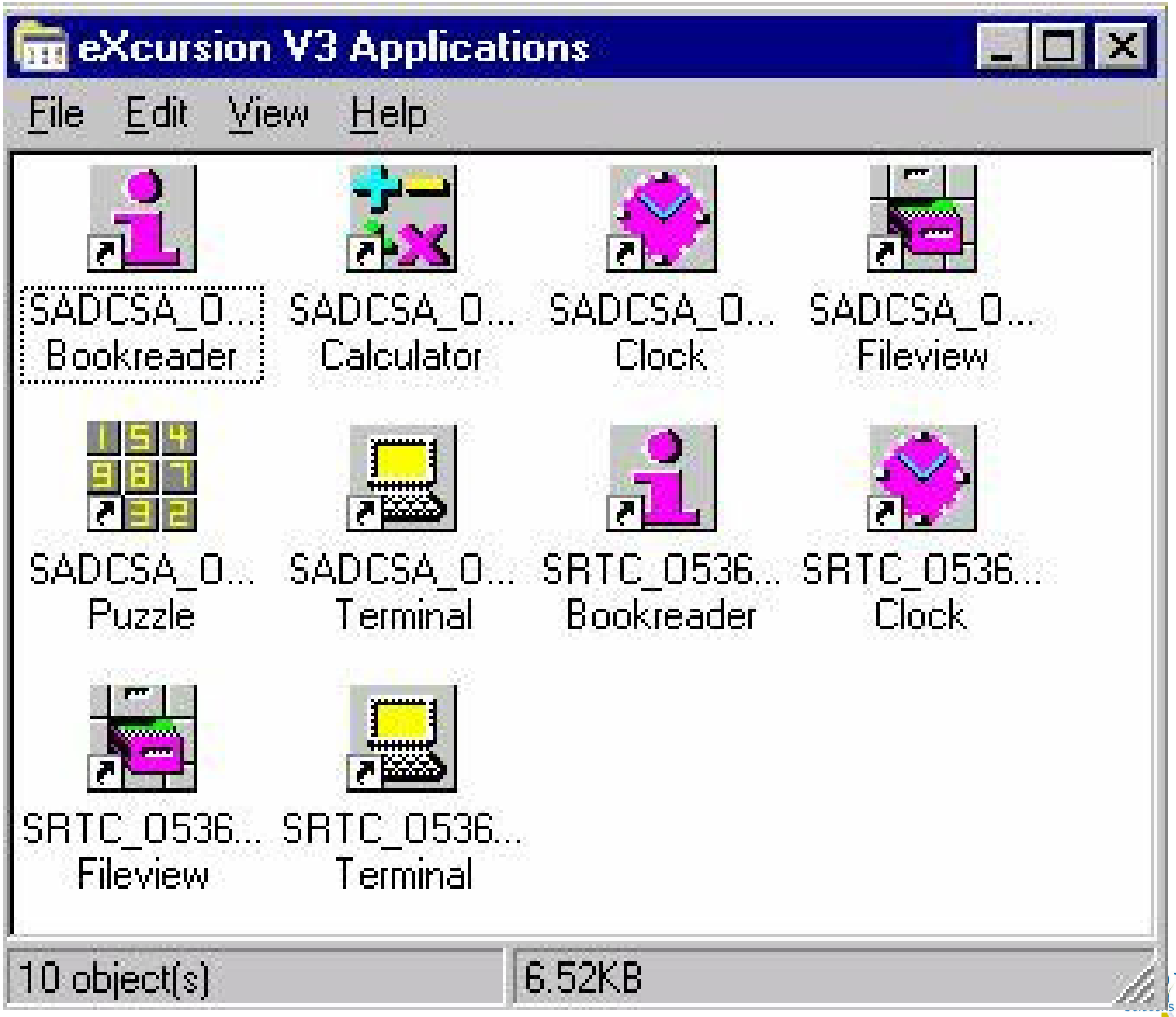
- Run the setup.exe file in the eXcursion directory

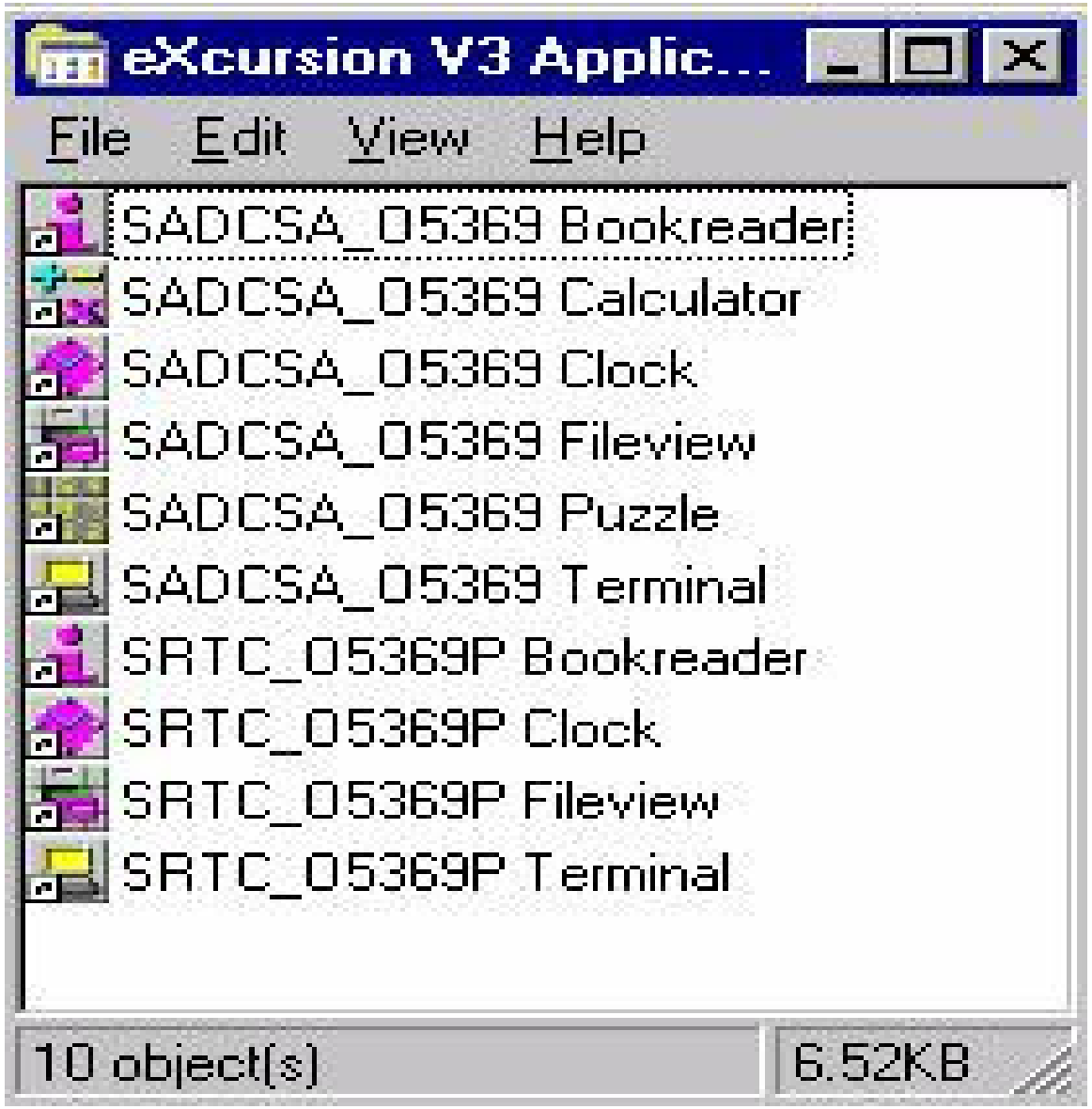



eXcursion Control Panel
Logging
Access
XDMCP
Extensions
Sounds
Info
Accounts
Applications
XWindow
Display
Fonts
Keyboard
Mouse
eXcursion Control Panel
Copyright © 1995, 1996
Digital Equipment Corporation
Computer Name: ARTHUR-COCHRANE
Registered to:
Serial Number:
Installed Path: C:\WIN32APP\XCURSION\x86\
Version: V3.0.571
X Server
Server Number: 0
Number of Links in Use: 2
Number of Errors Logged: 0
Restart Server
OK
Cancel
Help


eXcursion Control Panel
Logging
Access
XDMCP
Extensions
Sounds
Info
Accounts
Applications
XWindow
Display
Fonts
Keyboard
Mouse
List of Account Aliases:
SADCSA_05369
SRTC_05369P
Account Information
Account Alias:
SADCSA_05369
Host:
sadcsa.srs.gov
Username:
o5369
Prompt for Username
Password:
Prompt for Password
Network Transport:
TCP/IP (rexec)
Command Interpreter (Shell):
OpenVMS/DCL
Add
Modify
Remove
Reset
Restart Server
OK
Cancel
Help


eXcursion Control Panel
Logging
Access
XDMCP
Extensions
Sounds
Info
Accounts
Applications
XWindow
Display
Fonts
Keyboard
Mouse
List of Application Aliases:
SADCSA_05369 Bookreader
SADCSA_05369 Calculator
SADCSA_05369 Clock
SADCSA_05369 Fileview
SADCSA_05369 Puzzle
SADCSA_05369 Terminal
SRTC_05369P Bookreader
SRTC_05369P Clock
SRTC_05369P Fileview
SRTC_05369P Terminal
Application Information
Application Alias:
SADCSA_05369 Bookreader
Command:
sys$system:decw$bookreader
Account Alias:
SADCSA_05369
Run at Server Startup
Prompt for Parameters
Add Icon to Program Group
Add Icon
Add
Remove
Modify
Run
Reset
Pre-Defined X-Applications
Click this button to select pre-defined X-applications
X-Applications ...
Restart Server
OK
Cancel
Help


eXcursion V3 Applications
File Edit View Help
SADCSA_0...
Bookreader
SADCSA_0...
Calculator
SADCSA_0...
Clock
SADCSA_0...
Fileview
SADCSA_0...
Puzzle
SADCSA_0...
Terminal
SRTC_0536...
Bookreader
SRTC_0536...
Clock
SRTC_0536...
Fileview
SRTC_0536...
Terminal
10 object(s)
6.52KB


eXcursion V3 Applic...
File Edit View Help
SADCSA_05369 Bookreader
SADCSA_05369 Calculator
SADCSA_05369 Clock
SADCSA_05369 Fileview
SADCSA_05369 Puzzle
SADCSA_05369 Terminal
SRTC_05369P Bookreader
SRTC_05369P Clock
SRTC_05369P Fileview
SRTC_05369P Terminal
10 object(s)
6.52KB

Display AMDS Information

- When logged on interactivity set the X-windows display to a X-windows server

```
$ set_display== "set display/create-
      /transport=tcpip-
      /node=''f$getdvi("TT:","TT_ACCPORNAM")'"-"["-"]”
```

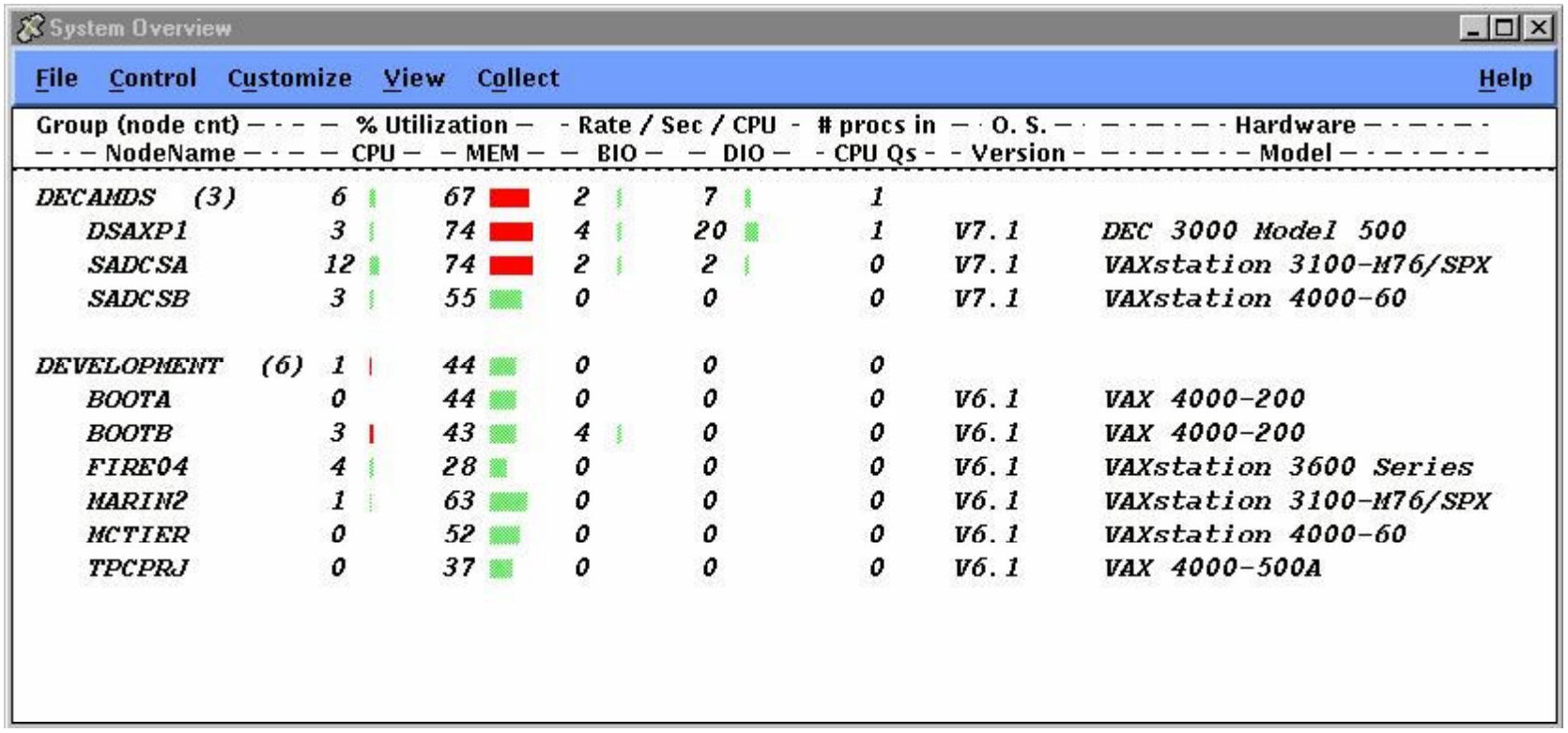
- Multinet does a set display on telnet, rlogin, and rshell access (unless disabled)
- Then enter the DCL command AVAIL/MOTIF

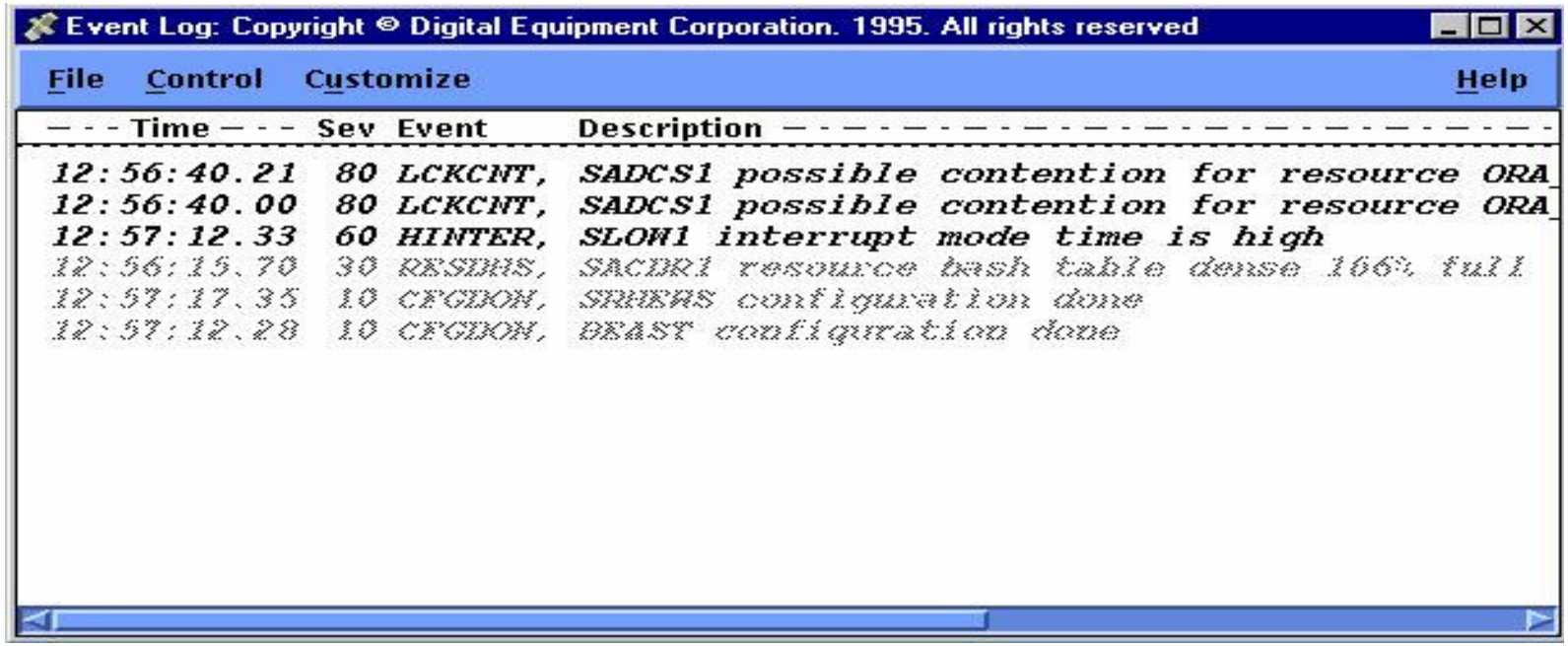
Server Site Files for Remote Login

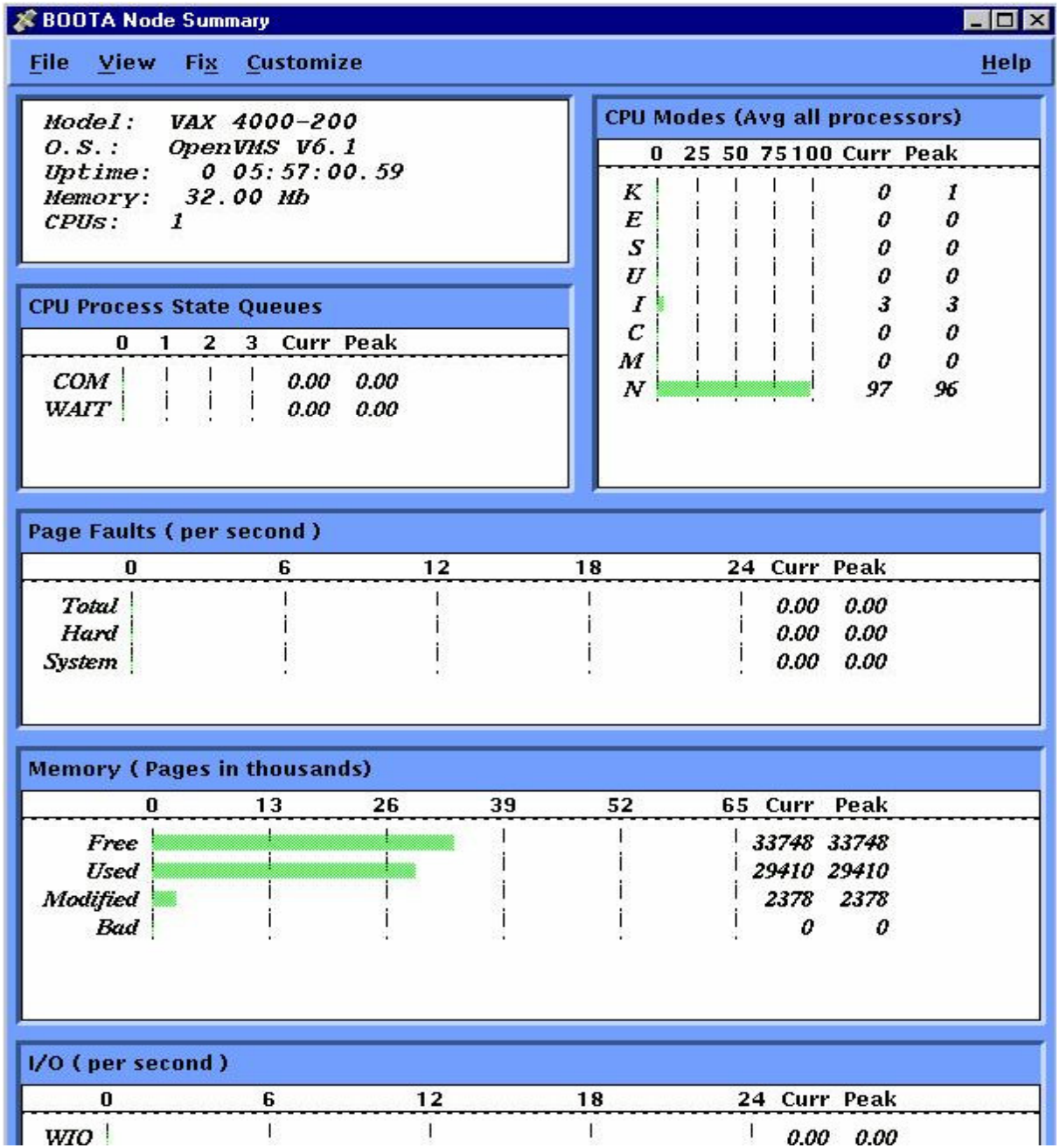
- If you have Pathworks installed then the files are already there
- If you do not have Pathworks then the following files need to be in sys$system and set world readable
  - `pcx$server.com`
  - `pcx$chkpwdexp.exe`
- eXcursion will do a REXEC login to the VMS host and execute the pcx$server.com file and cause it to execute the requested X-windows application to be run and displayed on the PC

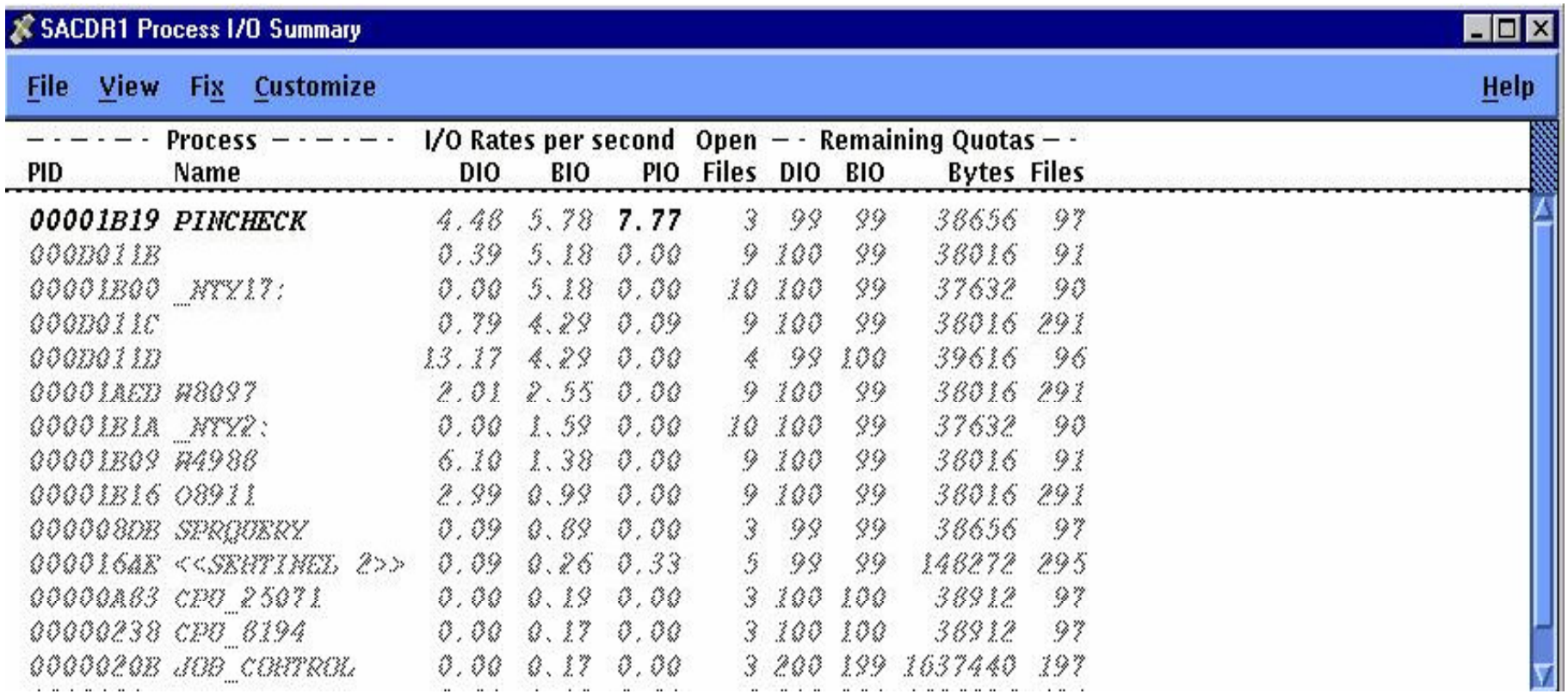
AMDS

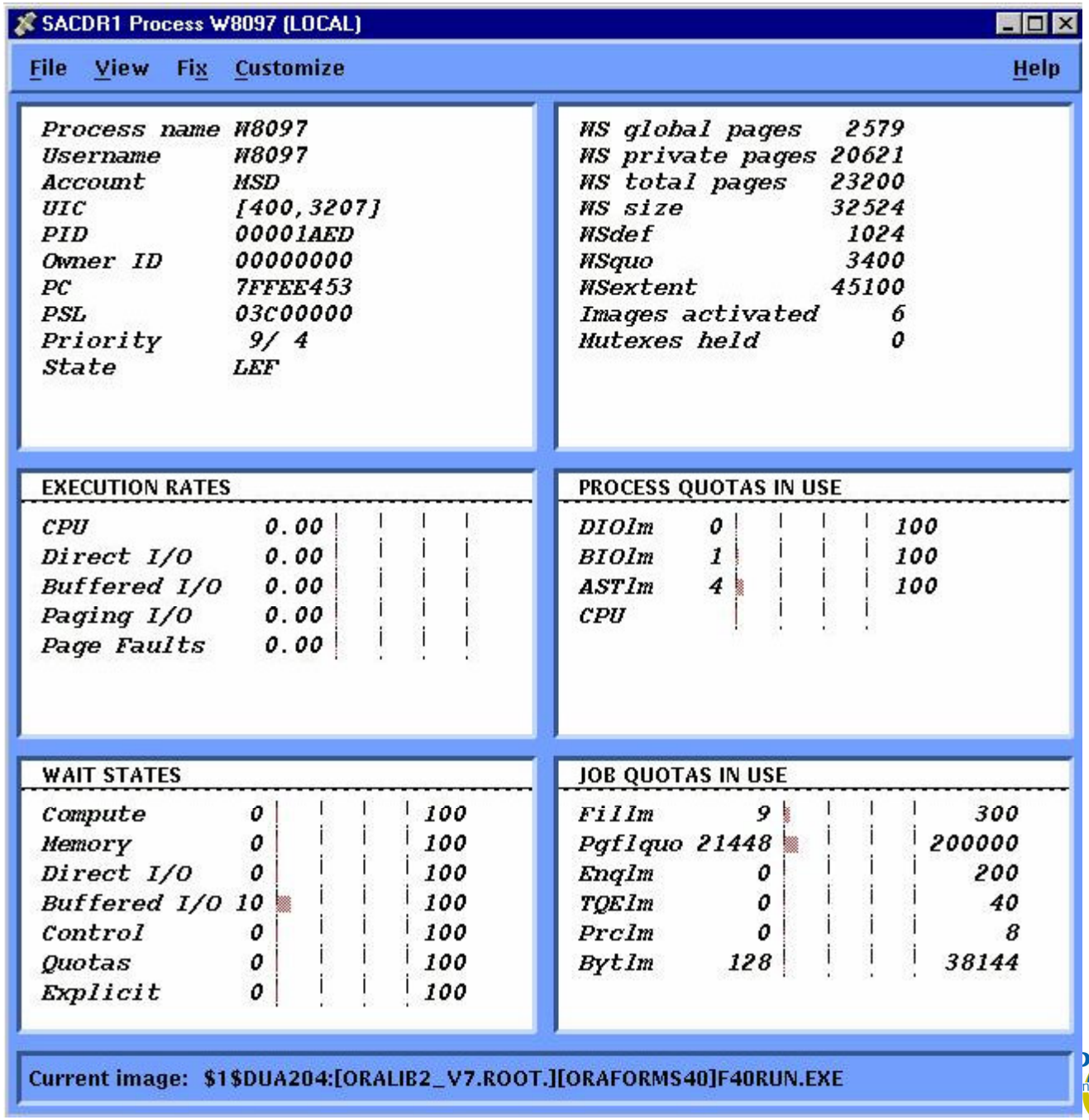
- Now here are some AMDS displays
- When viewing a display almost anything can be double clicked to drill down for more information
- AMDS can force quorum be recalculated in a hung cluster!

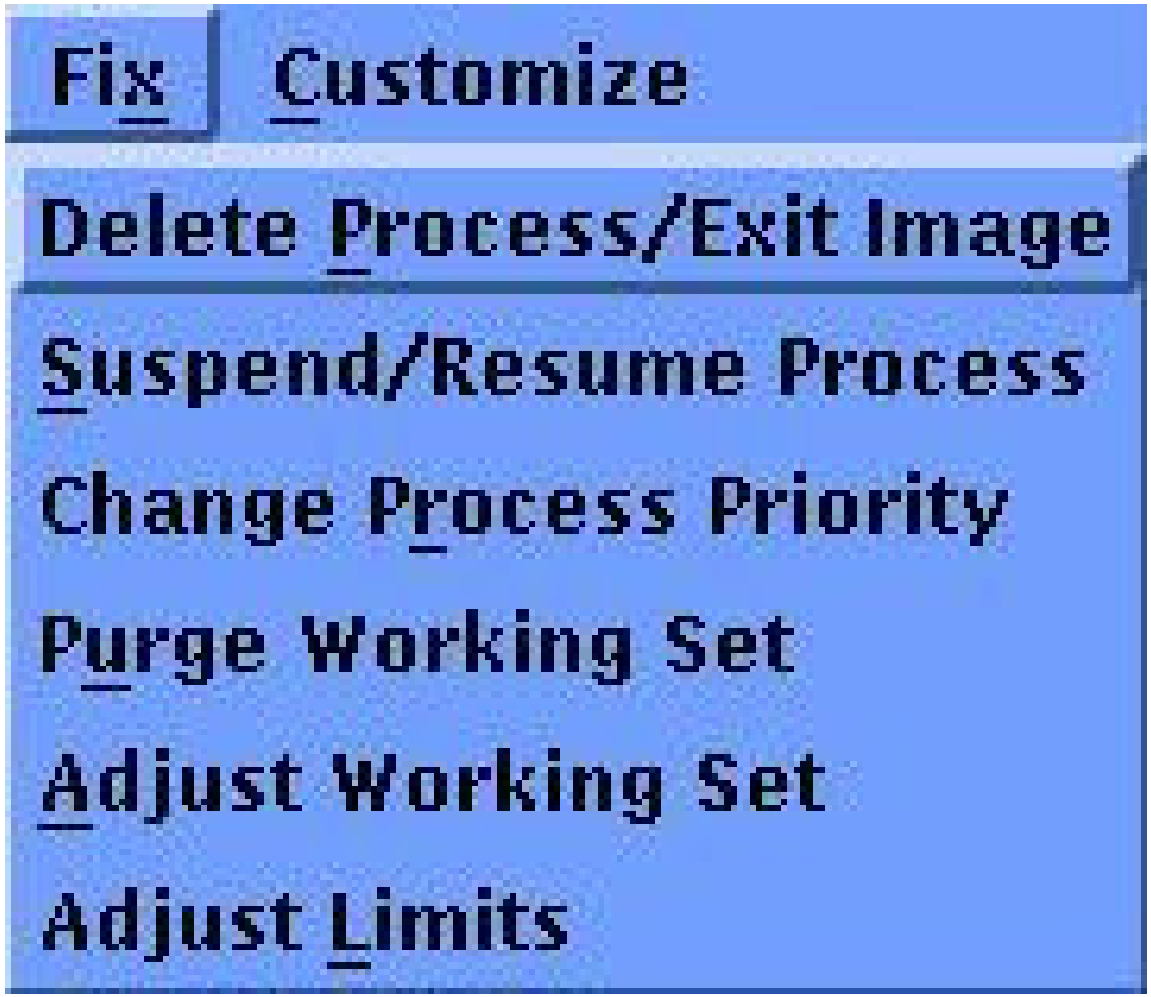

System Overview
File Control Customize View Collect Help
Group (node cnt) % Utilization Rate / Sec / CPU # procs in O. S. Hardware
NodeName CPU MEM BIO DIO CPU Qs Version Model
DECAMDS (3) 6 67 2 7 1
DSAXP1 3 74 4 20 1 V7.1 DEC 3000 Model 500
SADCSA 12 74 2 2 0 V7.1 VAXstation 3100-M76/SPX
SADCSB 3 55 0 0 0 V7.1 VAXstation 4000-60
DEVELOPMENT (6) 1 44 0 0 0
BOOTA 0 44 0 0 0 V6.1 VAX 4000-200
BOOTB 3 43 4 0 0 V6.1 VAX 4000-200
FIRE04 4 28 0 0 0 V6.1 VAXstation 3600 Series
MARIN2 1 63 0 0 0 V6.1 VAXstation 3100-M76/SPX
MCTIER 0 52 0 0 0 V6.1 VAXstation 4000-60
TPCPRJ 0 37 0 0 0 V6.1 VAX 4000-500A

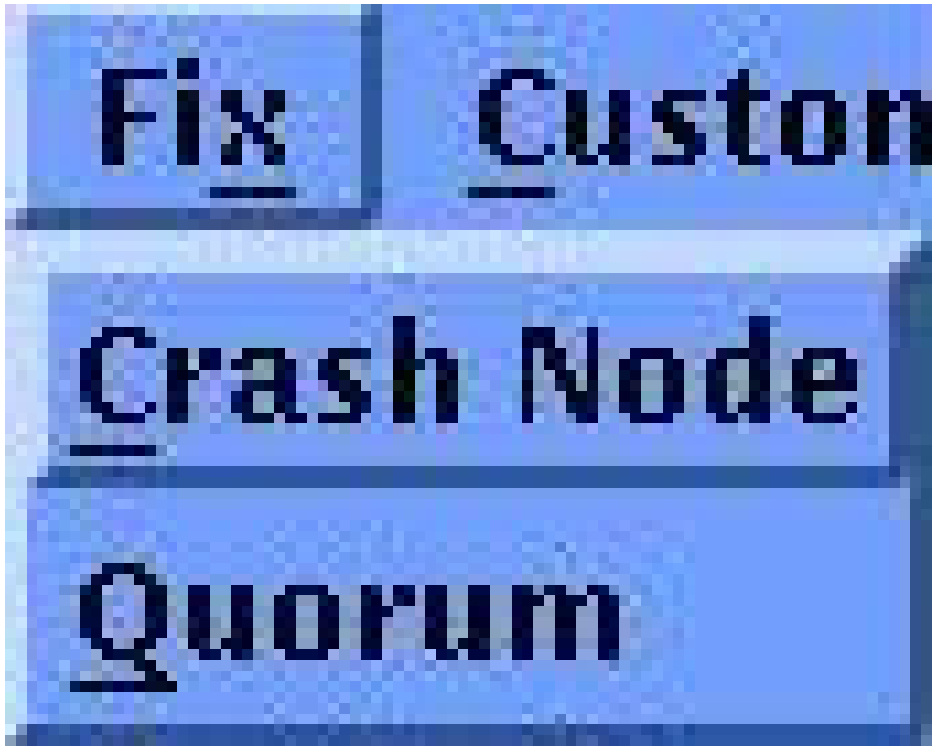

Event Log: Copyright © Digital Equipment Corporation. 1995. All rights reserved
File Control Customize Help
Time Sev Event Description
12:56:40.21 80 LCKCNT, SADCS1 possible contention for resource ORA
12:56:40.00 80 LCKCNT, SADCS1 possible contention for resource ORA
12:57:12.33 60 HINTER, SLOW1 interrupt mode time is high
12:56:15.70 30 RSSDNS, SACDR1 resource hash table dense 106% full
12:57:17.35 10 CFGDON, SRVRMS configuration done
12:57:12.28 10 CFGDON, DSAST configuration done


BOOTA Node Summary
File View Fix Customize Help
Model: VAX 4000-200
O.S.: OpenVMS V6.1
Uptime: 0 05:57:00.59
Memory: 32.00 Mb
CPUs: 1
CPU Process State Queues
0 1 2 3 Curr Peak
COM 0.00 0.00
WAIT 0.00 0.00
CPU Modes (Avg all processors)
0 25 50 75 100 Curr Peak
K 0 1
E 0 0
S 0 0
U 0 0
I 3 3
C 0 0
M 0 0
N 97 96
Page Faults (per second)
0 6 12 18 24 Curr Peak
Total 0.00 0.00
Hard 0.00 0.00
System 0.00 0.00
Memory (Pages in thousands)
0 13 26 39 52 65 Curr Peak
Free 33748 33748
Used 29410 29410
Modified 2378 2378
Bad 0 0
I/O (per second)
0 6 12 18 24 Curr Peak
WIO 0.00 0.00


SACDR1 Process I/O Summary
File View Fix Customize Help
Process I/O Rates per second Open Remaining Quotas
PID Name DIO BIO PIO Files DIO BIO Bytes Files
00001B19 PINCHECK 4.48 5.78 7.77 3 99 99 38656 97
000D011B 0.39 5.18 0.00 9 100 99 38016 91
00001B00 _NTY17: 0.00 5.18 0.00 10 100 99 37632 90
000D011C 0.79 4.29 0.09 9 100 99 38016 291
000D011D 13.17 4.29 0.00 4 99 100 39616 96
00001AED R8097 2.01 2.55 0.00 9 100 99 38016 291
00001B1A _NTY2: 0.00 1.59 0.00 10 100 99 37632 90
00001B09 R4986 6.10 1.38 0.00 9 100 99 38016 91
00001B16 08911 2.99 0.99 0.00 9 100 99 38016 291
00000BDB SPRQUERY 0.09 0.69 0.00 3 99 99 38656 97
00001BAE <<SENTINEL 2>> 0.09 0.26 0.33 5 99 99 148272 295
00000A83 CPU_25071 0.00 0.19 0.00 3 100 100 38912 97
00000238 CPU_8194 0.00 0.17 0.00 3 100 100 38912 97
00000208 JOB_CONTROL 0.00 0.17 0.00 3 200 199 1637440 197


SACDR1 Process W8097 (LOCAL)
File View Fix Customize Help
Process name W8097
Username W8097
Account MSD
UIC [400,3207]
PID 00001AED
Owner ID 00000000
PC 7FFEE453
PSL 03C00000
Priority 9/ 4
State LEF
WS global pages 2579
WS private pages 20621
WS total pages 23200
WS size 32524
WSdef 1024
WSquo 3400
WSextent 45100
Images activated 6
Mutexes held 0
EXECUTION RATES
CPU 0.00
Direct I/O 0.00
Buffered I/O 0.00
Paging I/O 0.00
Page Faults 0.00
PROCESS QUOTAS IN USE
DIOlm 0 100
BIOlm 1 100
ASTlm 4 100
CPU
WAIT STATES
Compute 0 100
Memory 0 100
Direct I/O 0 100
Buffered I/O 10 100
Control 0 100
Quotas 0 100
Explicit 0 100
JOB QUOTAS IN USE
Fillm 9 300
Pgflquo 21448 200000
Enqlm 0 200
TQElm 0 40
Prclm 0 8
Bytlm 128 38144
Current image: 1DUA204:[ORALIB2_V7.ROOT.][ORAFORMS40]F40RUN.EXE


Fix
Customize
Delete Process/Exit Image
Suspend/Resume Process
Change Process Priority
Purge Working Set
Adjust Working Set
Adjust Limits


Fix
Custom
Crash Node
Quorum

OpenVMS Management Tools

- PC client and OpenVMS server for user, printer management, and storage management

OpenVMS Server Software

- OpenVMS 6.2 has v1.0-B
- OpenVMS 7.1 has v2.1
- Can install v2.1 on OpenVMS 6.2
  - can be installed on OpenVMS 6.1 with VAXCXXL01-061 installed
- Version 3.0A on OpenVMS 7.2-1
- Version 3.2 on OpenVMS 7.3-1
  - Version 3.x is based on the Microsoft Management Console
- Can be downloaded from the OpenVMS web site

Install and Startup

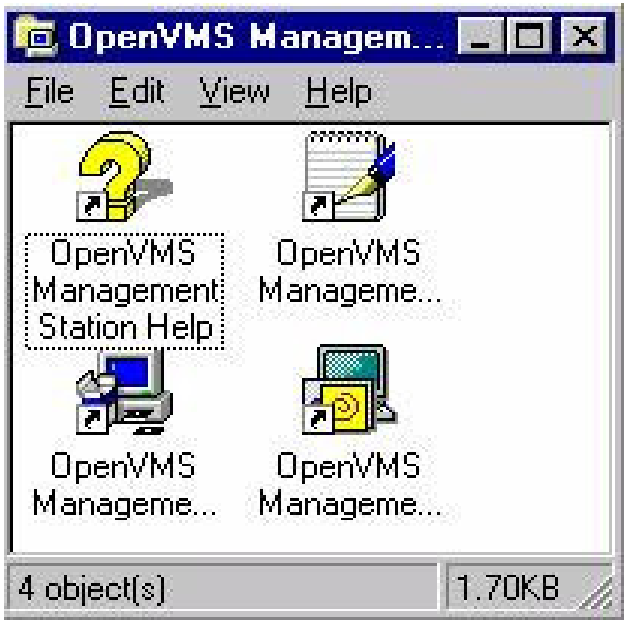
- Install with the PRODUCT command
- Add to SYSTARTUP_VMS.COM
  - `@SYS$STARTUP:TNT$STARTUP BOOT`
- Add to SYSHUTDWN.COM
  - `@SYS$STARTUP:TNT$SHUTDOWN`
- TNT_SERVER process gets created
- Now ready for PC client

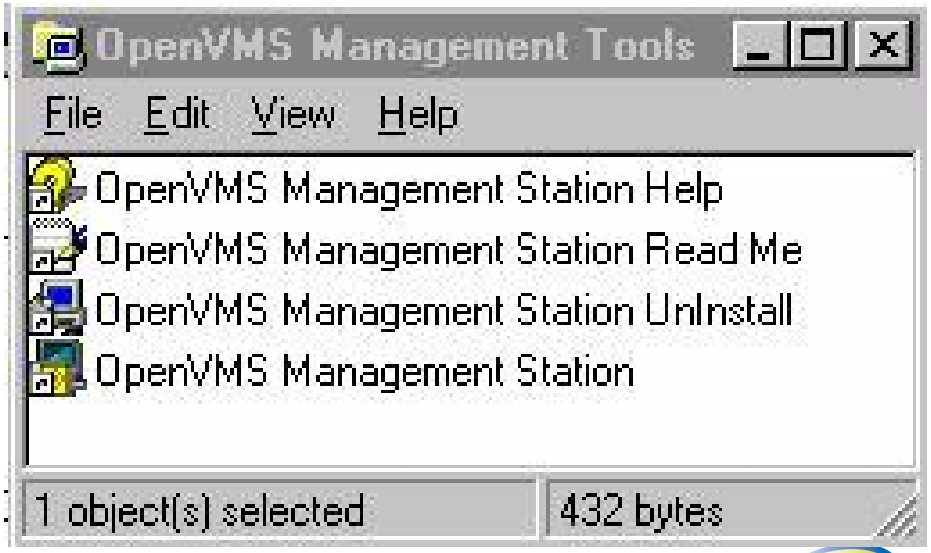
PC Client Software

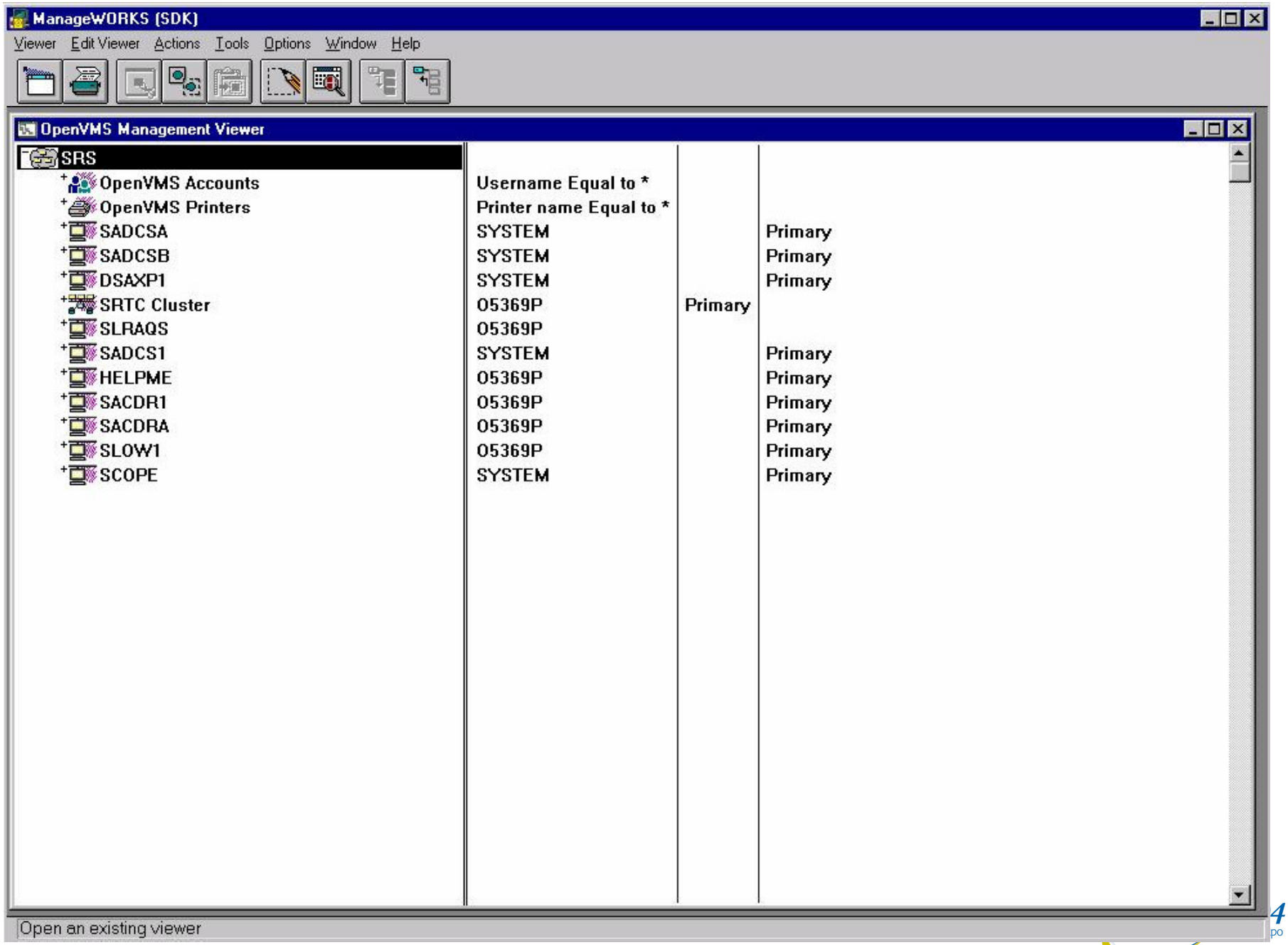
- PC client files are an option during a VMS install or upgrade
  - `SYS$COMMON:[TNT.CLIENT]TNT032.EXE  ! V7.3-1`
  - `SYS$COMMON:[TNT.CLIENT]DISKIMAG.EXE`
  - `SYS$COMMON:[TNT.CLIENT]TNTCLIDx.IMG`
    - `x=1 & 2 for v6.2 and x= 1 to 6 for v7.1`
    - `Note the .EXE file is an Intel PC executable`
- Transfer the files to the PC via FTP or Pathworks
- OpenVMS Management Tools CD that comes with the Alpha CONDIST
- Version 3.0 from the OpenVMS web site

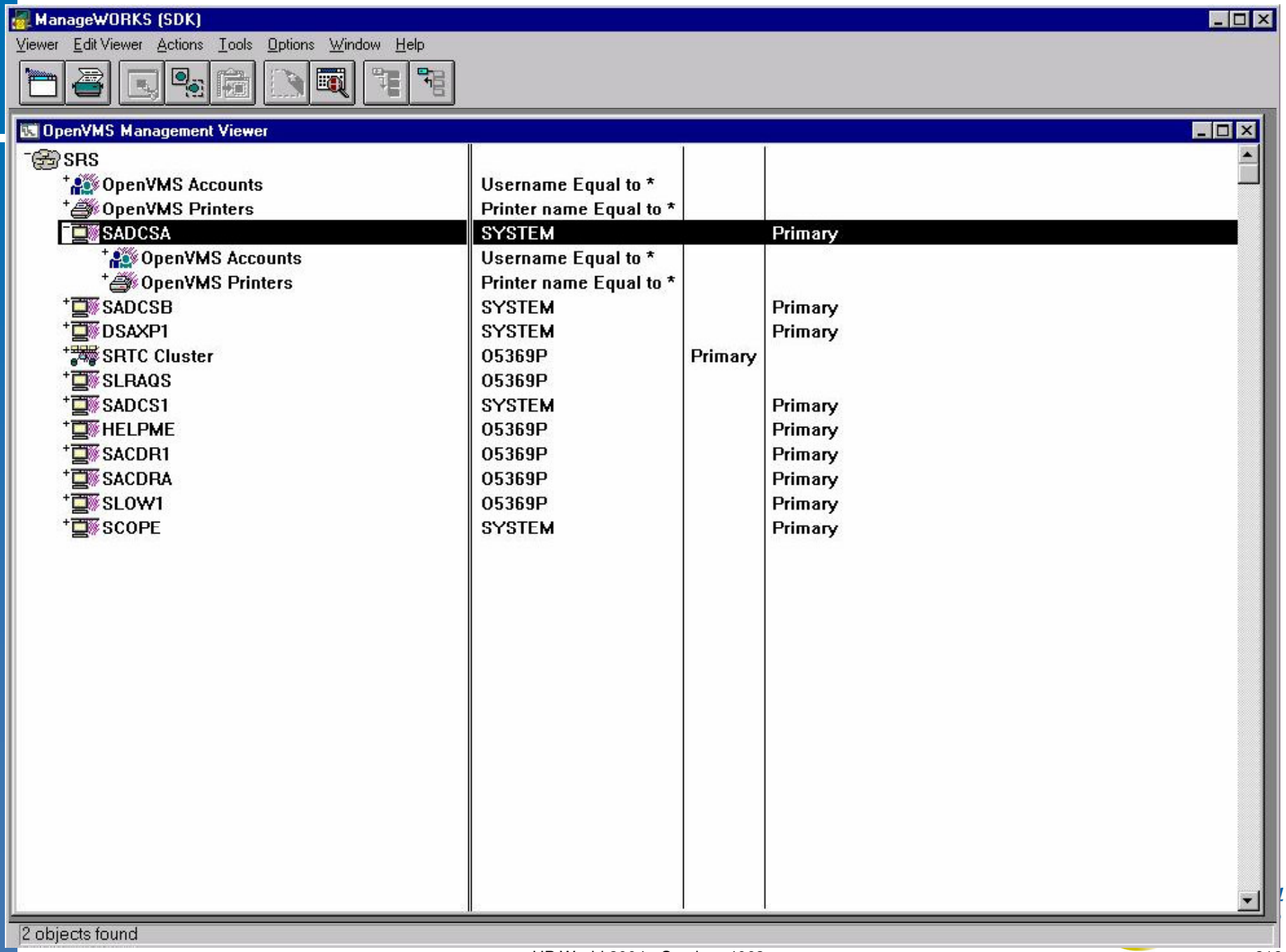
Install PC client files

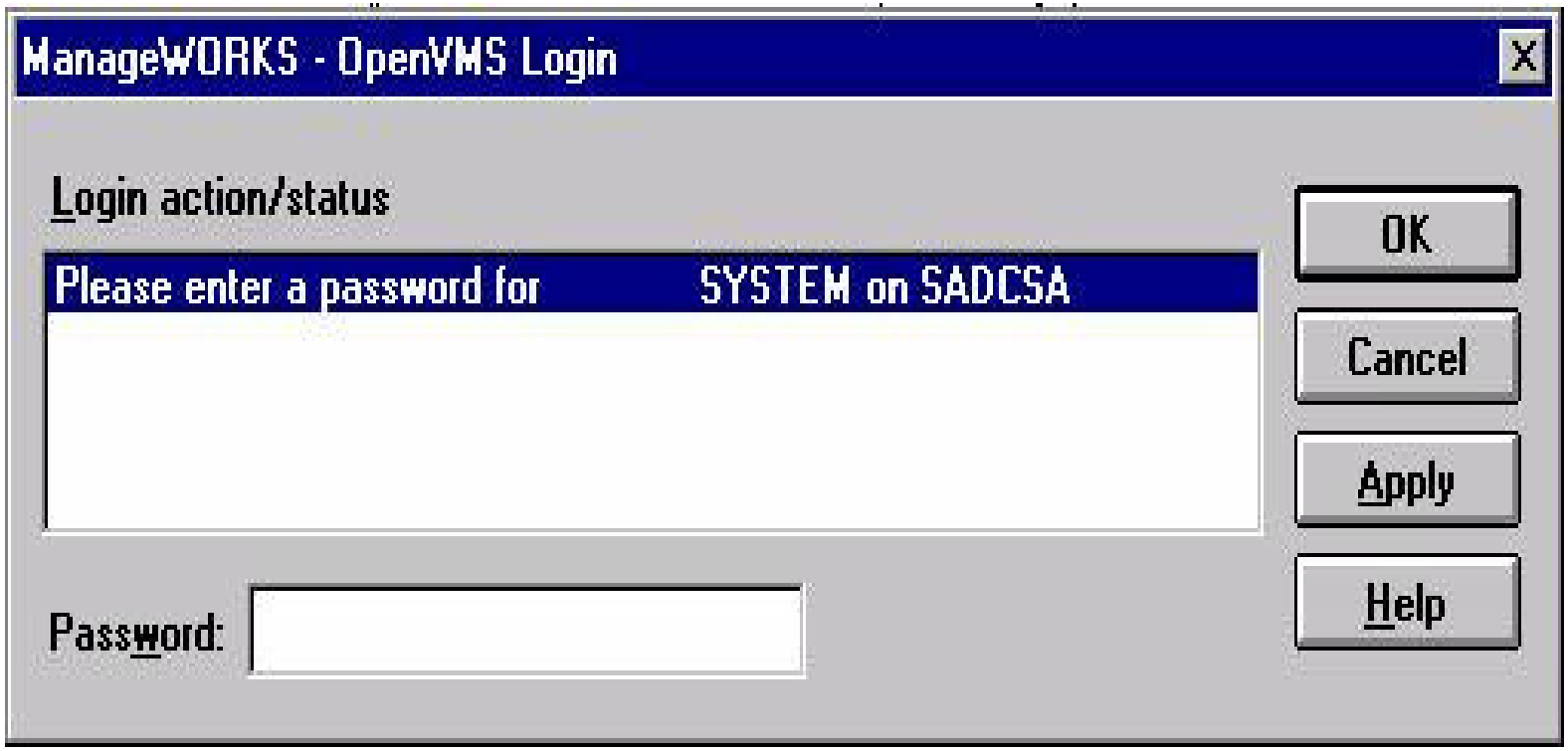
- Run the setup.exe file in the \tnt\disk1 directory
- A program group is created in the Start menu

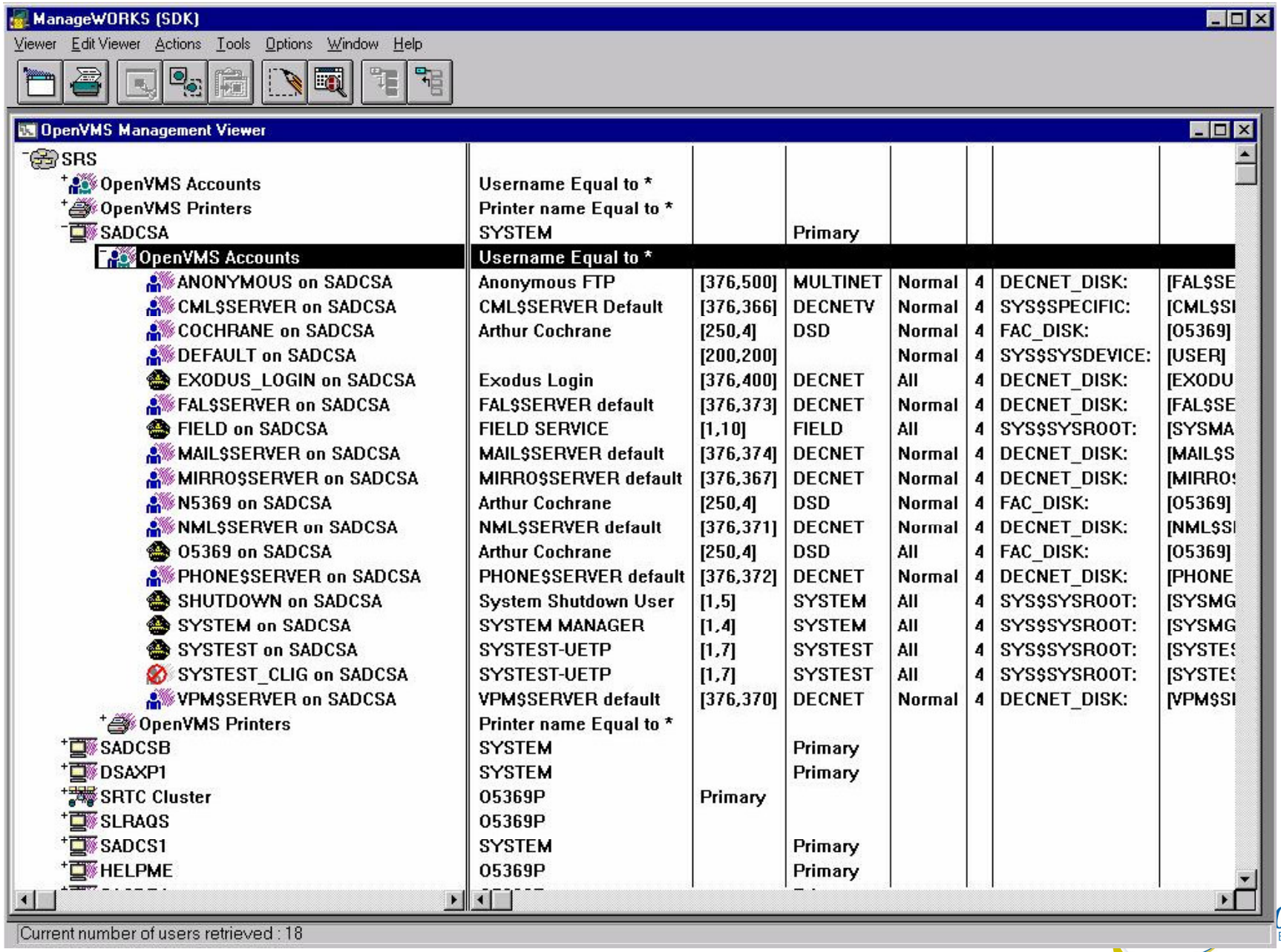


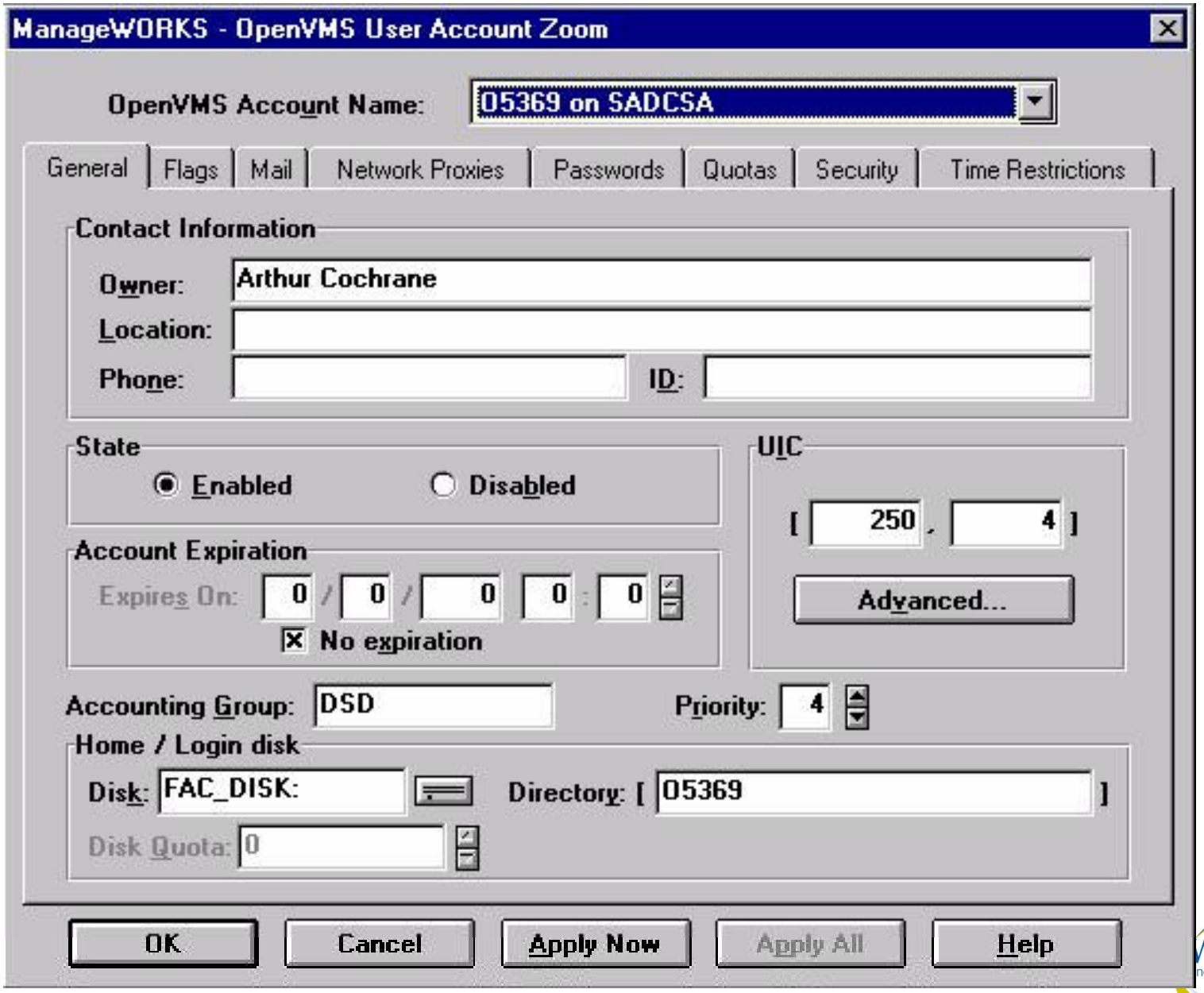


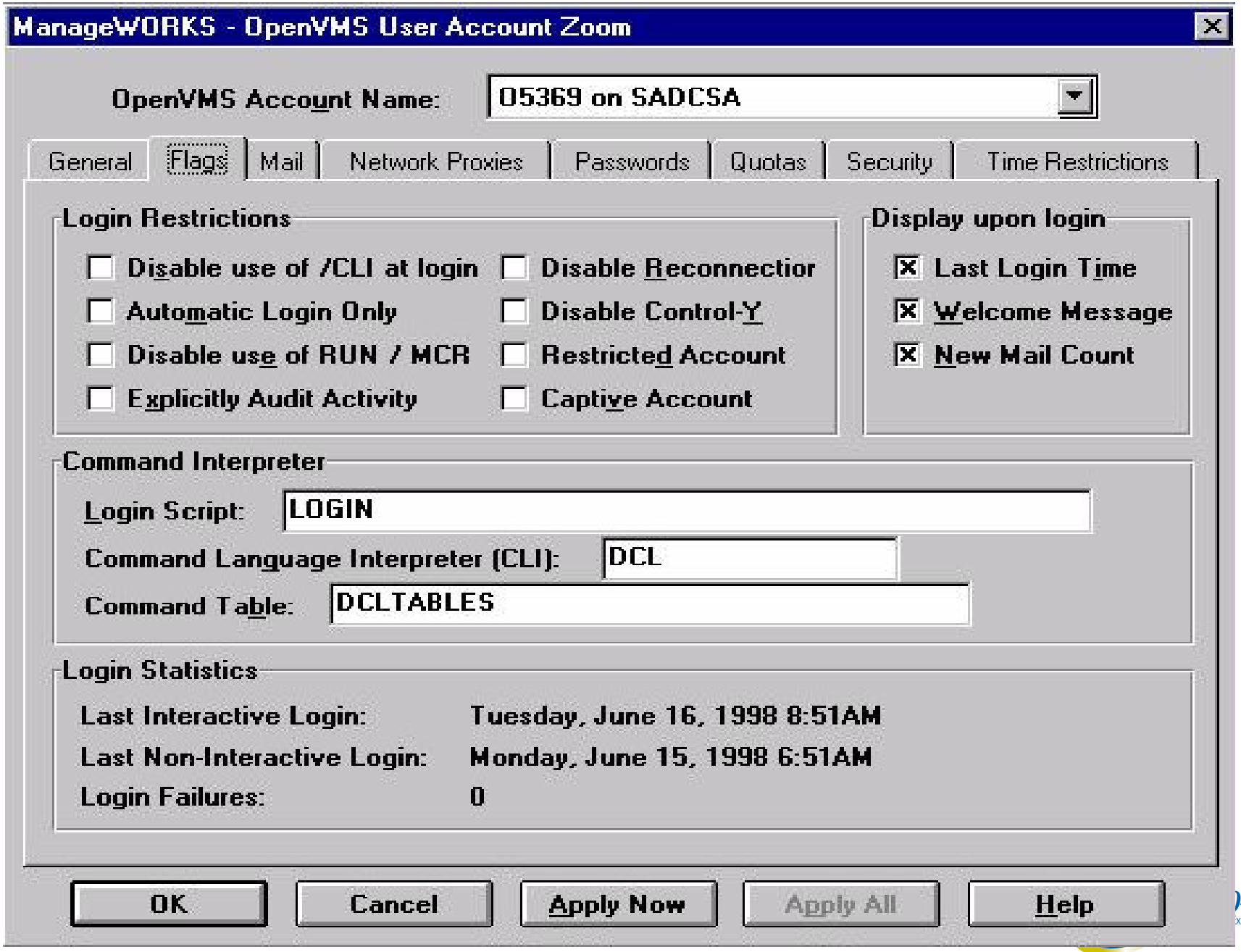

ManageWORKS (SDK)
Viewer Edit Viewer Actions Tools Options Window Help
OpenVMS Management Viewer
SRS
OpenVMS Accounts
OpenVMS Printers
SADCSA
SADCSB
DSAXP1
SRTC Cluster
SLRAQS
SADCS1
HELPME
SACDR1
SACDRA
SLOW1
SCOPE
Username Equal to *
Printer name Equal to *
SYSTEM
SYSTEM
SYSTEM
05369P
05369P
SYSTEM
05369P
05369P
05369P
05369P
SYSTEM
Primary
Primary
Primary
Primary
Primary
Primary
Primary
Primary
Primary
Open an existing viewer

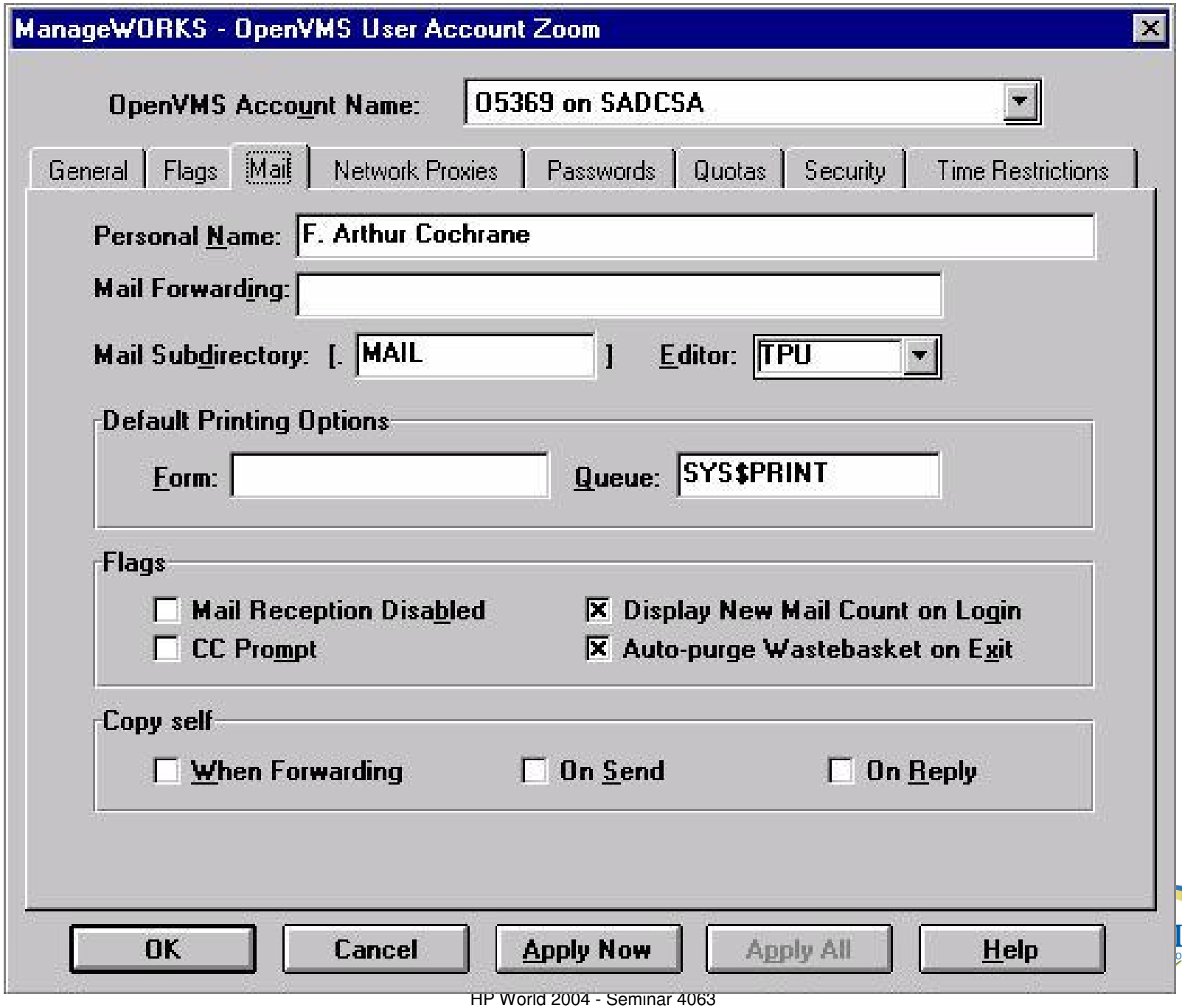

ManageWORKS (SDK)
Viewer Edit Viewer Actions Tools Options Window Help
OpenVMS Management Viewer
SRS
OpenVMS Accounts
Username Equal to *
OpenVMS Printers
Printer name Equal to *
SADCSA
SYSTEM
Primary
OpenVMS Accounts
Username Equal to *
OpenVMS Printers
Printer name Equal to *
SADCSB
SYSTEM
Primary
DSAXP1
SYSTEM
Primary
SRTC Cluster
O5369P
Primary
SLRAQS
O5369P
SADCS1
SYSTEM
Primary
HELPME
O5369P
Primary
SACDR1
O5369P
Primary
SACDRA
O5369P
Primary
SLOW1
O5369P
Primary
SCOPE
SYSTEM
Primary
2 objects found

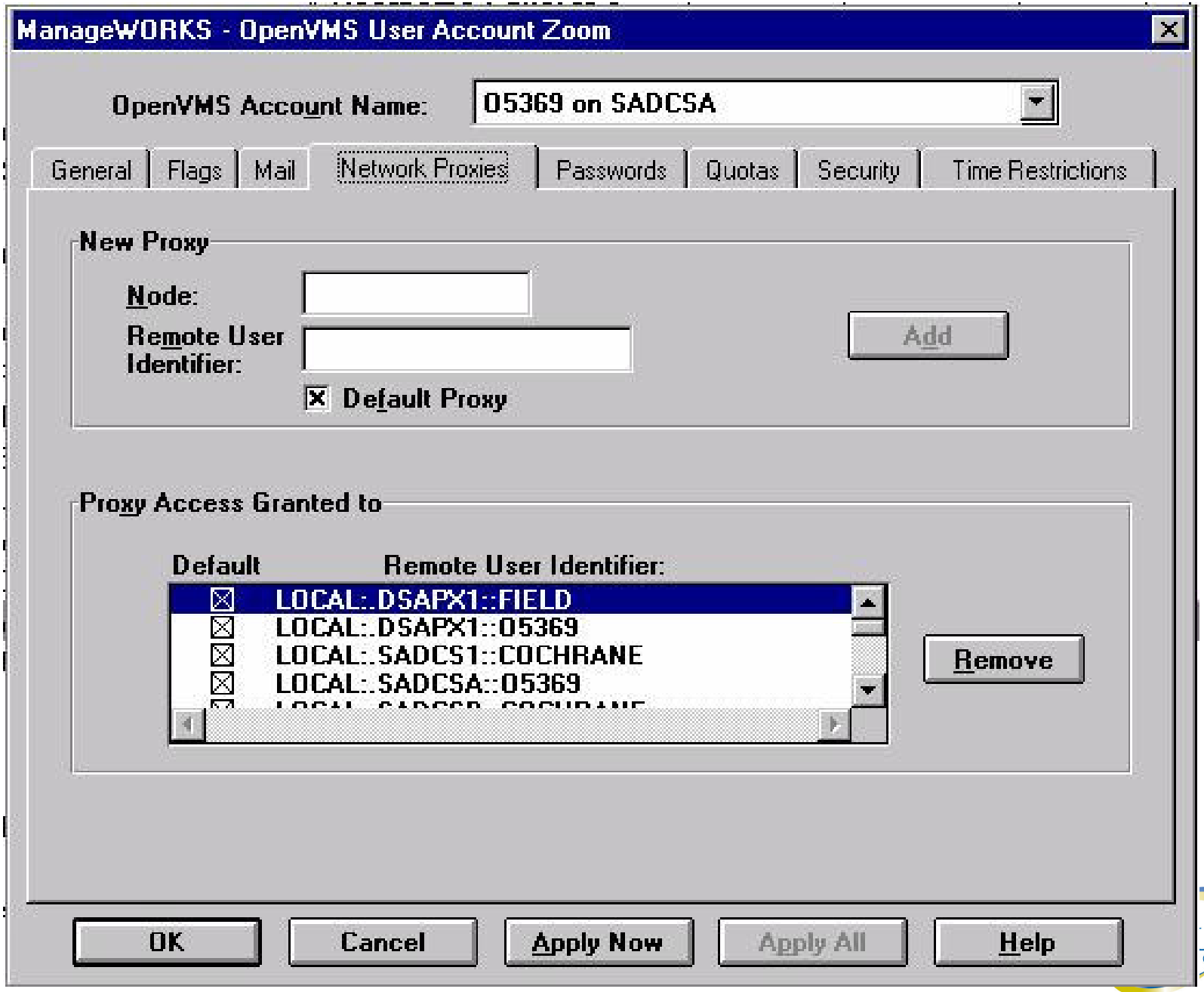

ManageWORKS - OpenVMS Login
Login action/status
Please enter a password for SYSTEM on SADCSA
Password:
OK
Cancel
Apply
Help

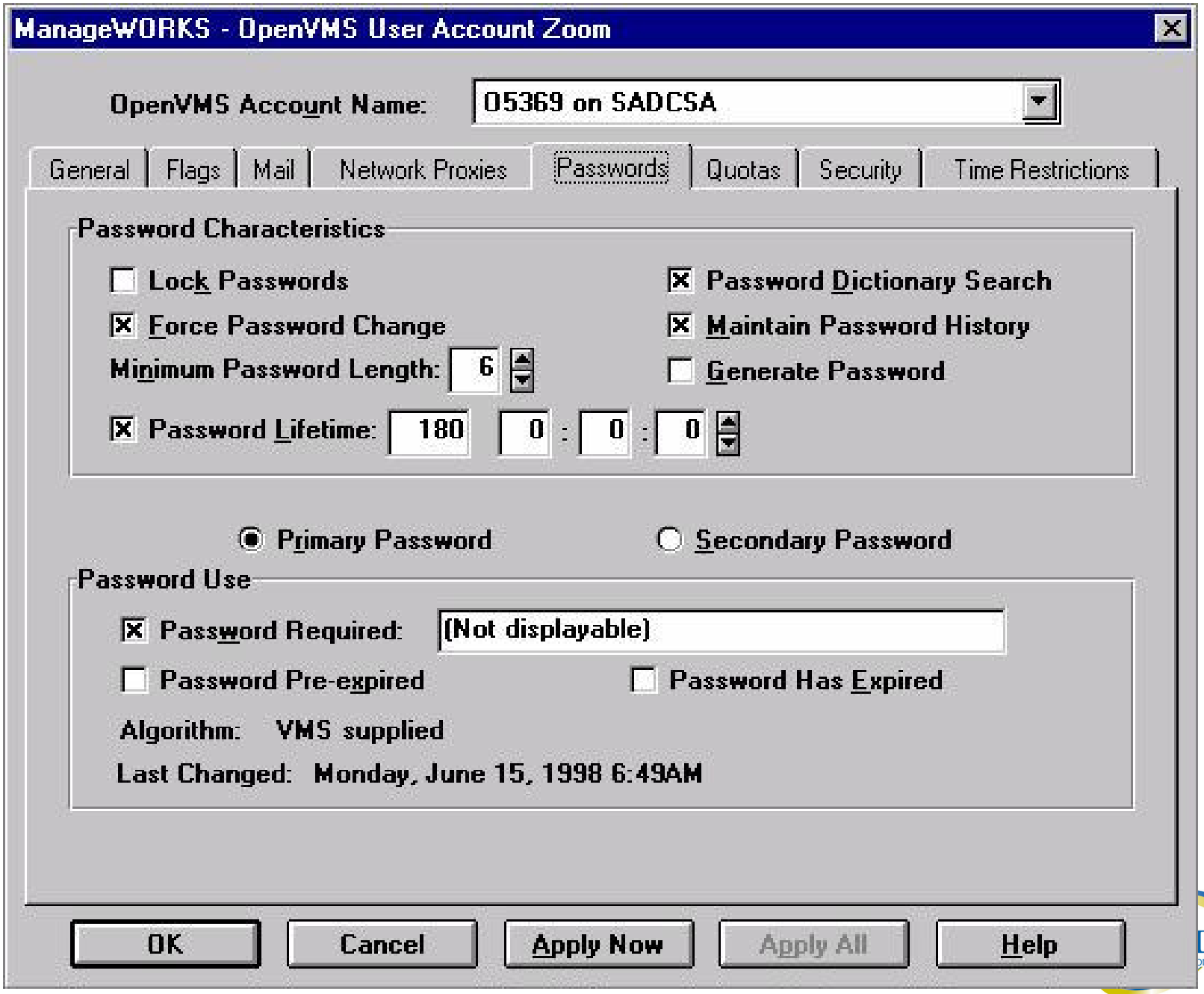

ManageWORKS (SDK)
Viewer Edit Viewer Actions Tools Options Window Help
OpenVMS Management Viewer
SRS
OpenVMS Accounts
OpenVMS Printers
SADCSA
OpenVMS Accounts
ANONYMOUS on SADCSA
CML$SERVER on SADCSA
COCHRANE on SADCSA
DEFAULT on SADCSA
EXODUS_LOGIN on SADCSA
FAL$SERVER on SADCSA
FIELD on SADCSA
MAIL$SERVER on SADCSA
MIRRO$SERVER on SADCSA
N5369 on SADCSA
NML$SERVER on SADCSA
O5369 on SADCSA
PHONE$SERVER on SADCSA
SHUTDOWN on SADCSA
SYSTEM on SADCSA
SYSTEST on SADCSA
SYSTEST_CLIG on SADCSA
VPM$SERVER on SADCSA
OpenVMS Printers
SADCSB
DSAXP1
SRTC Cluster
SLRAQS
SADCS1
HELPME
Username Equal to *
Printer name Equal to *
SYSTEM
Primary
Username Equal to *
Anonymous FTP
CML$SERVER Default
Arthur Cochrane
FIELD SERVICE
MAIL$SERVER default
MIRRO$SERVER default
NML$SERVER default
PHONE$SERVER default
System Shutdown User
SYSTEM MANAGER
SYSTEST-UETP
VPM$SERVER default
O5369P
Current number of users retrieved : 18

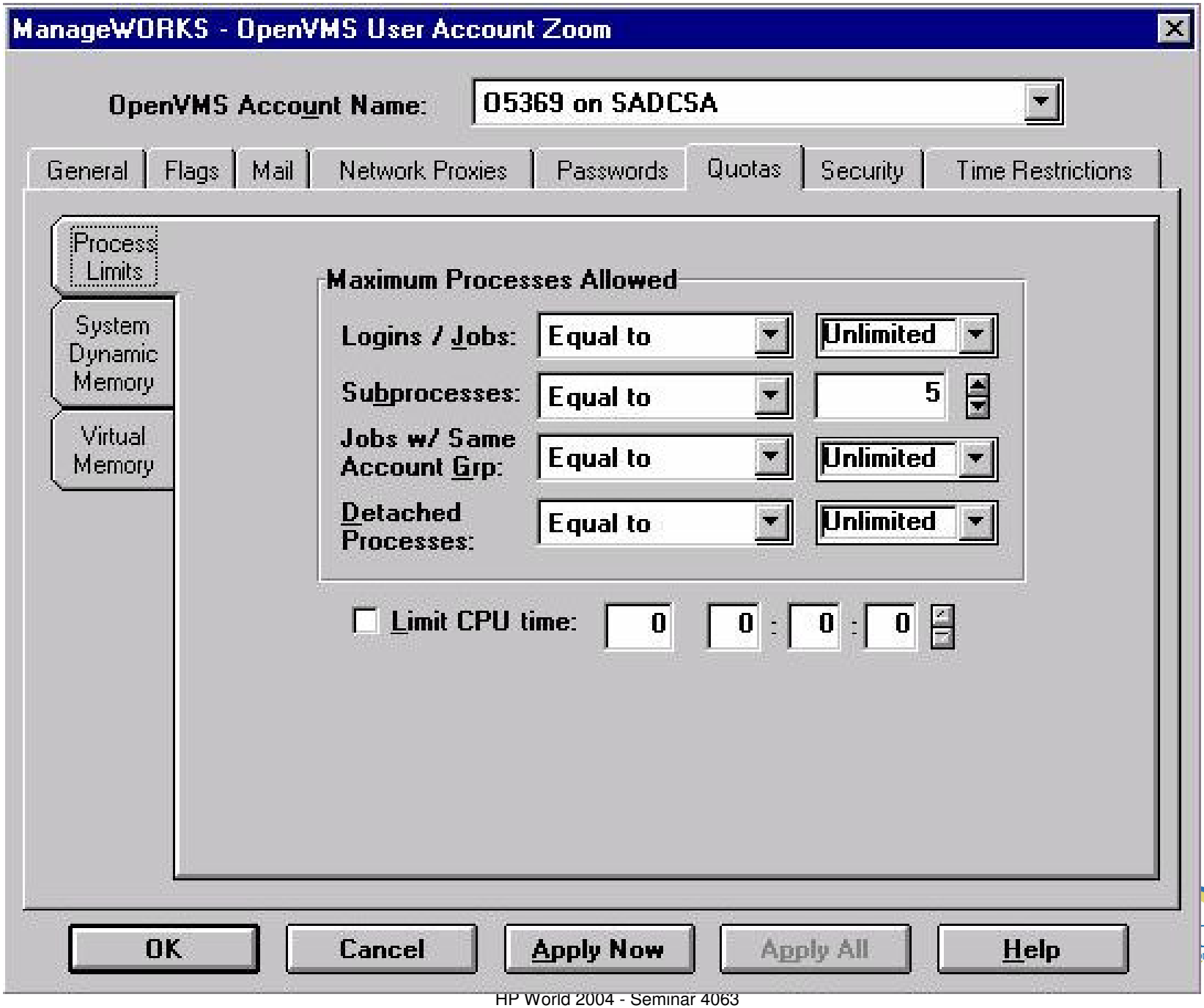

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Contact Information
Owner:
Arthur Cochrane
Location:
Phone:
ID:
State
Enabled
Disabled
UIC
[250 , 4]
Account Expiration
Expires On:
0 / 0 / 0 0 : 0
No expiration
Advanced...
Accounting Group:
DSD
Priority:
4
Home / Login disk
Disk:
FAC_DISK:
Directory: [05369]
Disk Quota:
0
OK
Cancel
Apply Now
Apply All
Help

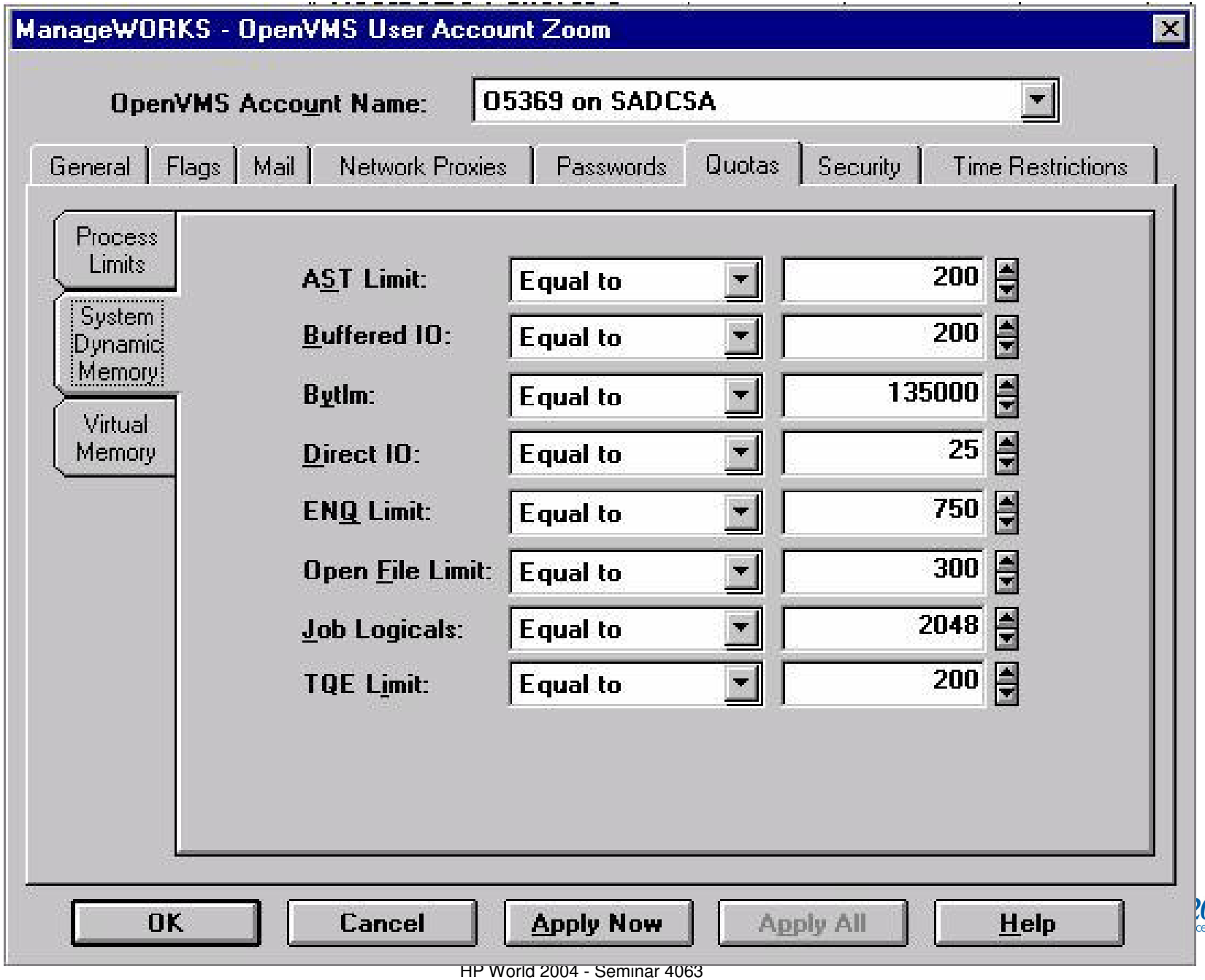

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Login Restrictions
Disable use of /CLI at login
Disable Reconnectior
Automatic Login Only
Disable Control-Y
Disable use of RUN / MCR
Restricted Account
Explicitly Audit Activity
Captive Account
Display upon login
Last Login Time
Welcome Message
New Mail Count
Command Interpreter
Login Script:
LOGIN
Command Language Interpreter (CLI):
DCL
Command Table:
DCLTABLES
Login Statistics
Last Interactive Login:
Tuesday, June 16, 1998 8:51AM
Last Non-Interactive Login:
Monday, June 15, 1998 6:51AM
Login Failures:
0
OK
Cancel
Apply Now
Apply All
Help

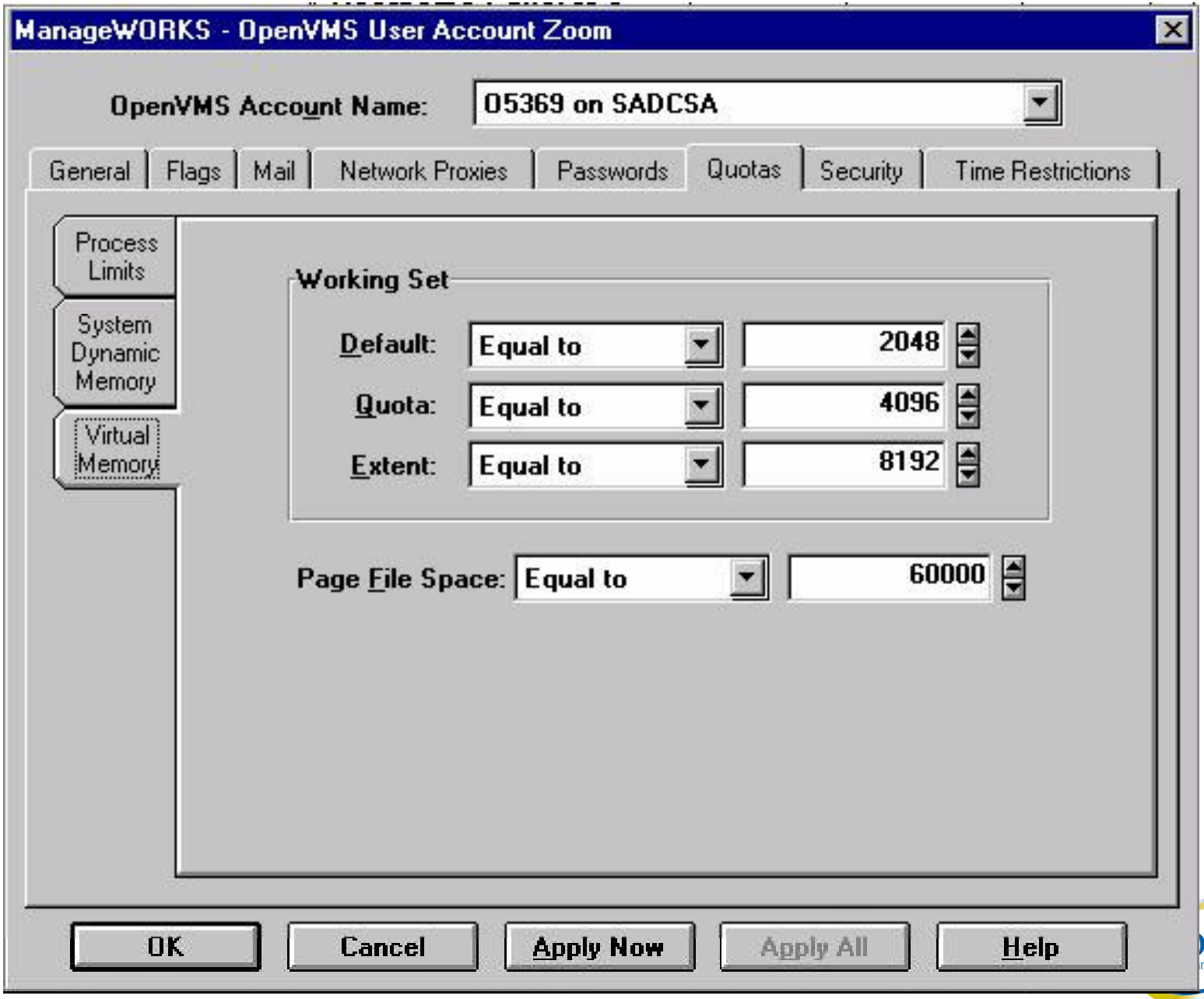

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Personal Name:
F. Arthur Cochrane
Mail Forwarding:
Mail Subdirectory: [.
MAIL
]
Editor:
TPU
Default Printing Options
Form:
Queue:
SYS$PRINT
Flags
Mail Reception Disabled
Display New Mail Count on Login
CC Prompt
Auto-purge Wastebasket on Exit
Copy self
When Forwarding
On Send
On Reply
OK
Cancel
Apply Now
Apply All
Help

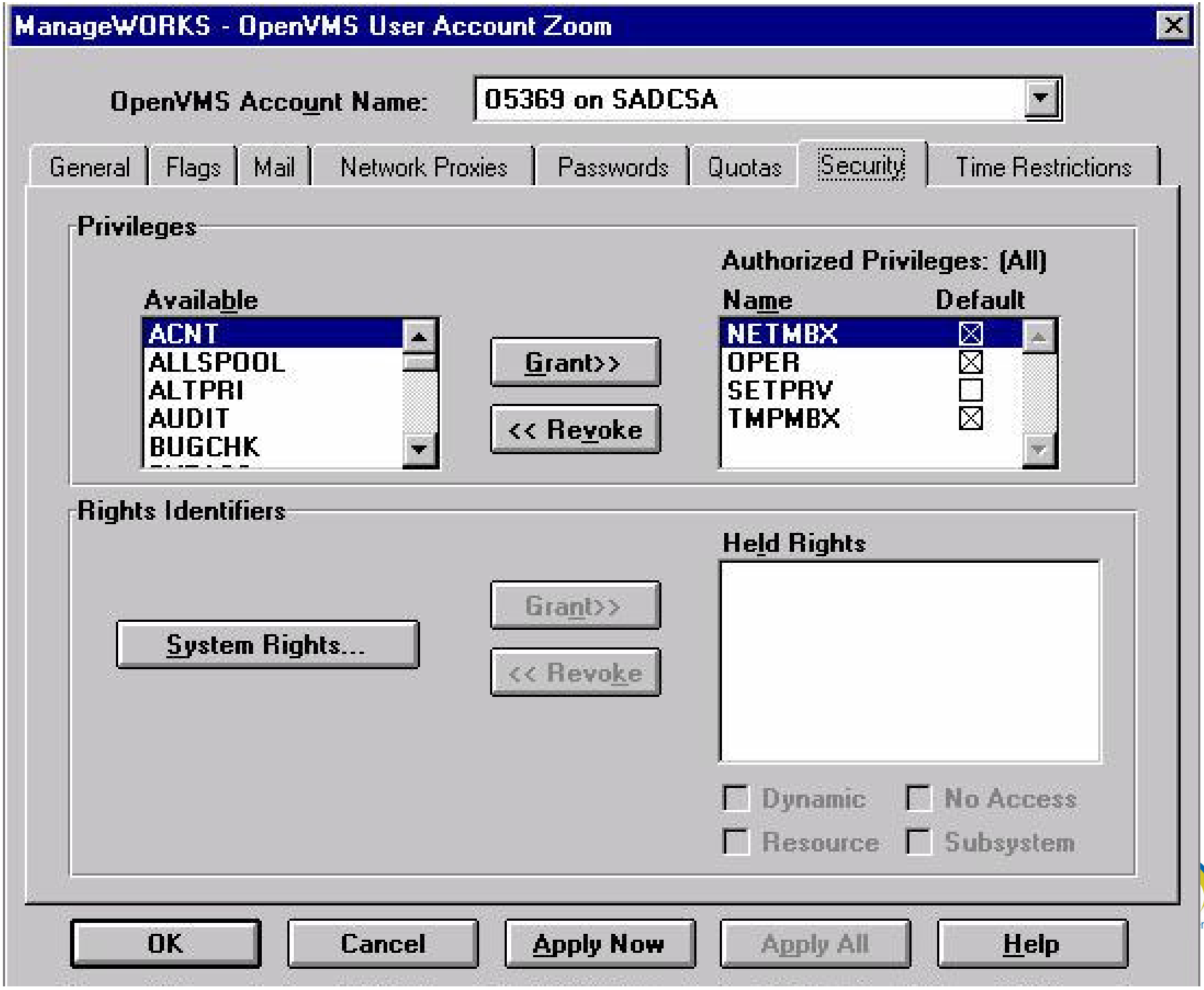

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
New Proxy
Node:
Remote User Identifier:
Add
Default Proxy
Proxy Access Granted to
Default
Remote User Identifier:
LOCAL:.DSAPX1::FIELD
LOCAL:.DSAPX1::05369
LOCAL:.SADCS1::COCHRANE
LOCAL:.SADCSA::05369
Remove
OK
Cancel
Apply Now
Apply All
Help

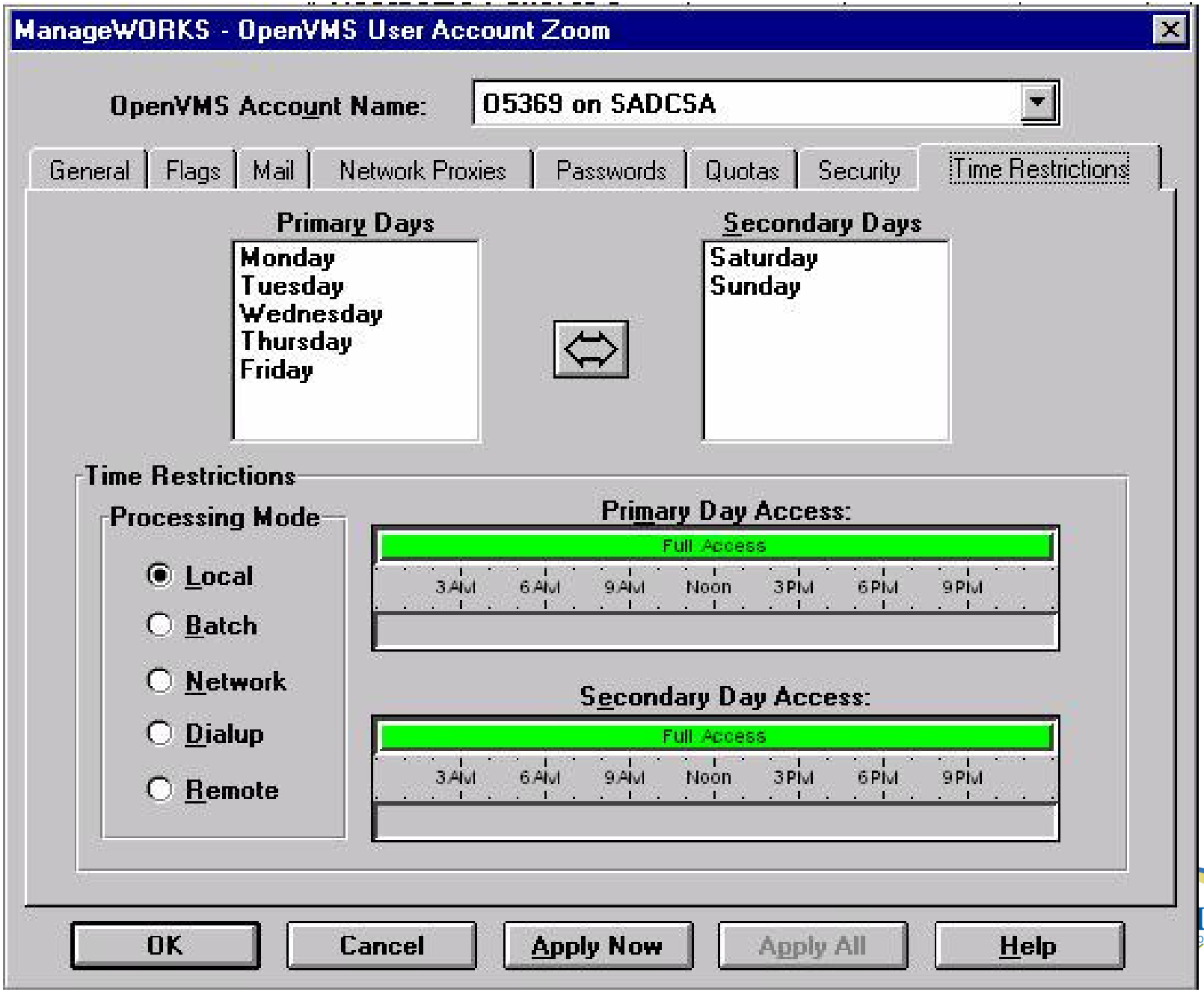

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Password Characteristics
Lock Passwords
Force Password Change
Minimum Password Length:
6
Password Lifetime:
180
0
0
0
Password Dictionary Search
Maintain Password History
Generate Password
Primary Password
Secondary Password
Password Use
Password Required:
(Not displayable)
Password Pre-expired
Password Has Expired
Algorithm: VMS supplied
Last Changed: Monday, June 15, 1998 6:49AM
OK
Cancel
Apply Now
Apply All
Help

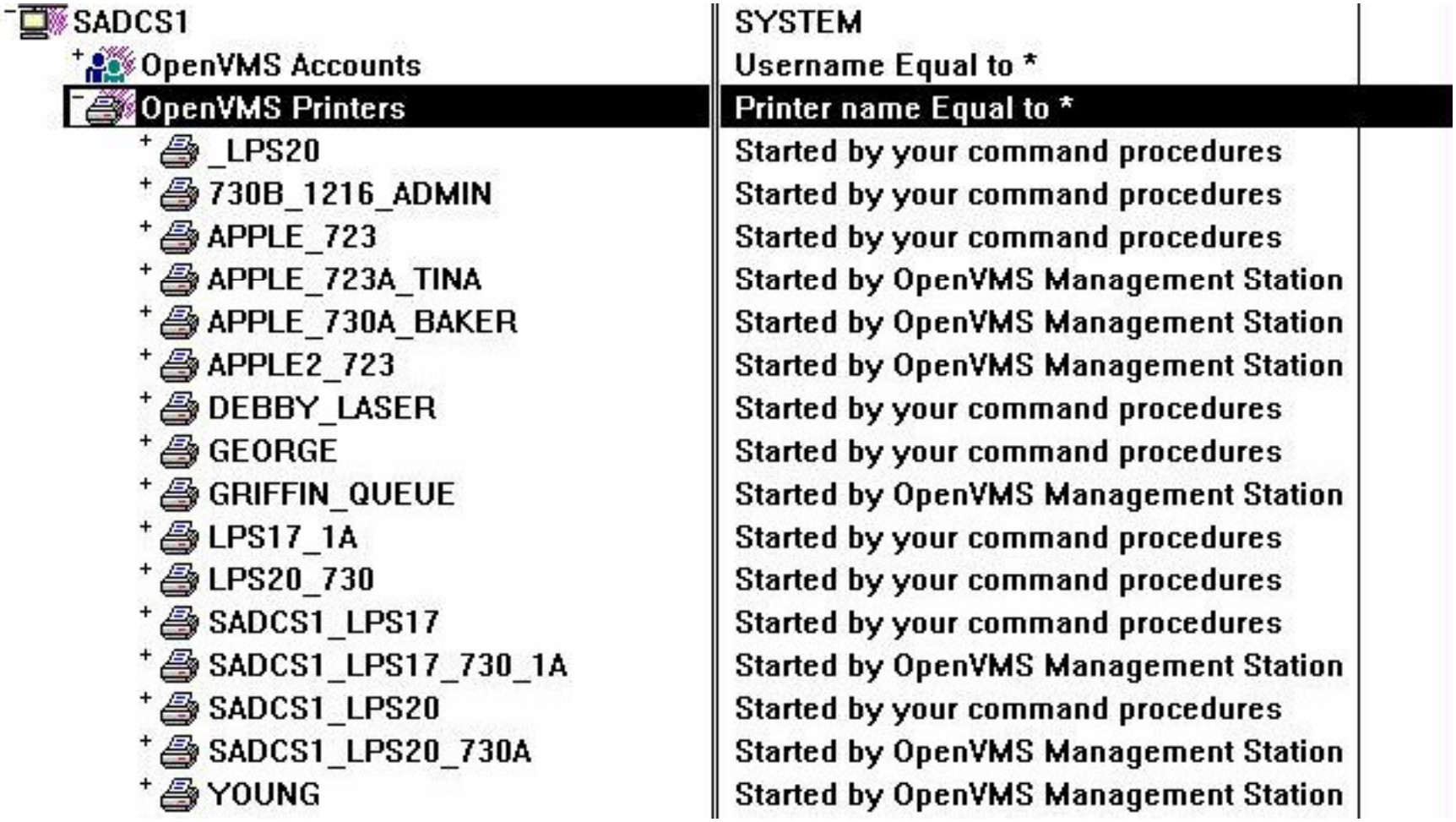

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Process Limits
System Dynamic Memory
Virtual Memory
Maximum Processes Allowed
Logins / Jobs:
Equal to
Unlimited
Subprocesses:
Equal to
5
Jobs w/ Same Account Grp:
Equal to
Unlimited
Detached Processes:
Equal to
Unlimited
Limit CPU time:
0
0 : 0 : 0
OK
Cancel
Apply Now
Apply All
Help

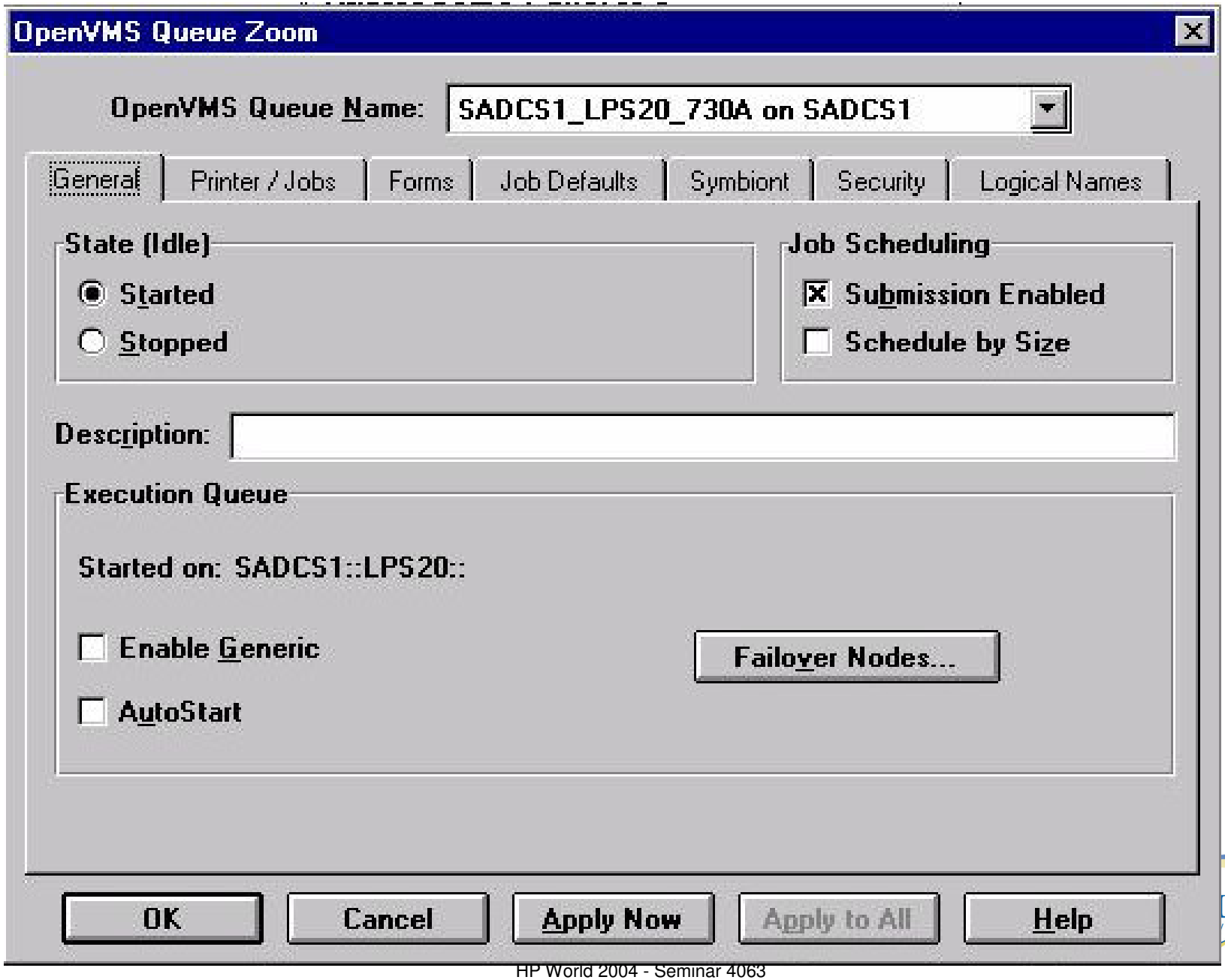

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Process Limits
System Dynamic Memory
Virtual Memory
AST Limit:
Equal to
200
Buffered IO:
Equal to
200
Bytlm:
Equal to
135000
Direct IO:
Equal to
25
ENQ Limit:
Equal to
750
Open File Limit:
Equal to
300
Job Logicals:
Equal to
2048
TQE Limit:
Equal to
200
OK
Cancel
Apply Now
Apply All
Help

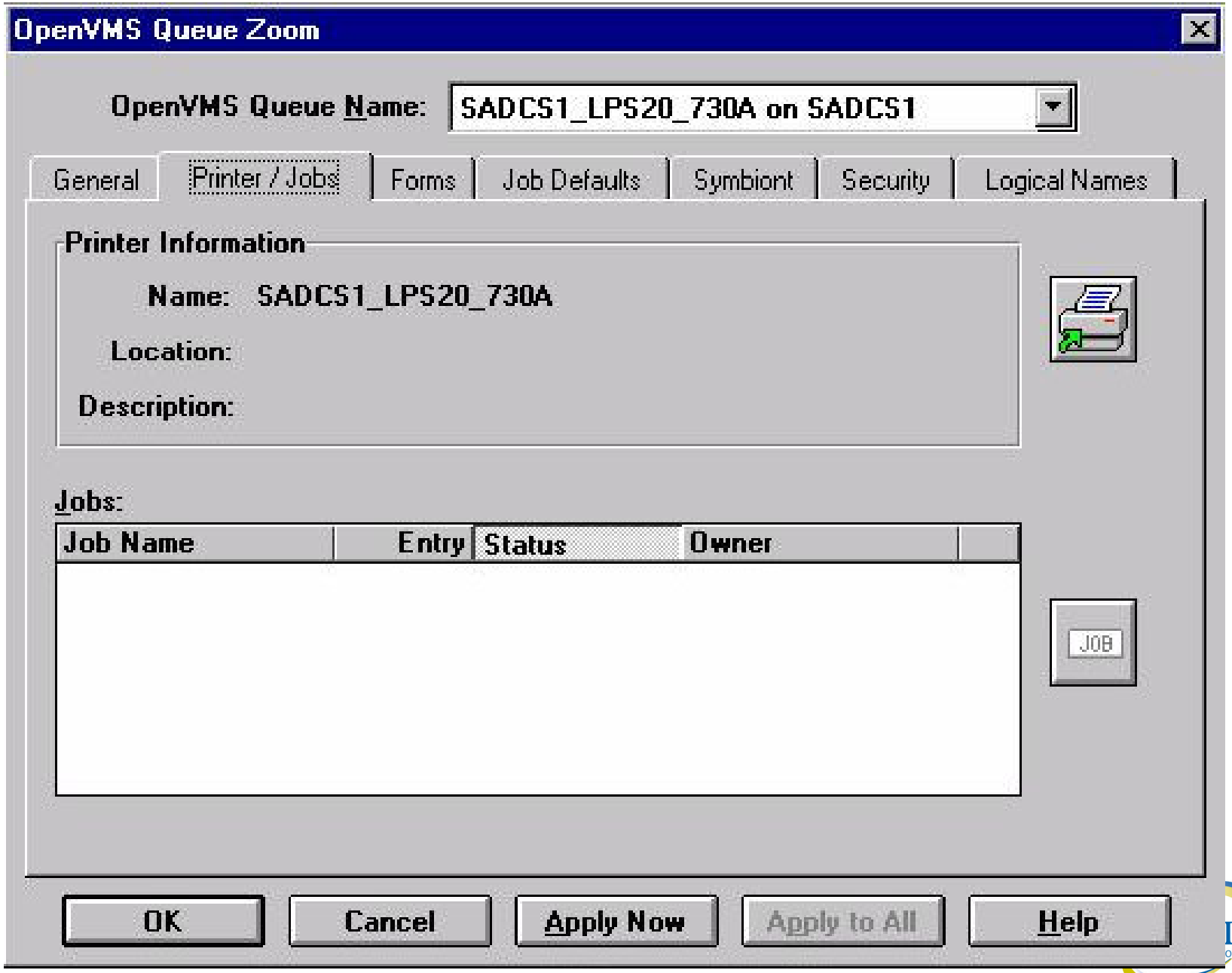

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Process Limits
System Dynamic Memory
Virtual Memory
Working Set
Default:
Equal to
2048
Quota:
Equal to
4096
Extent:
Equal to
8192
Page File Space:
Equal to
60000
OK
Cancel
Apply Now
Apply All
Help

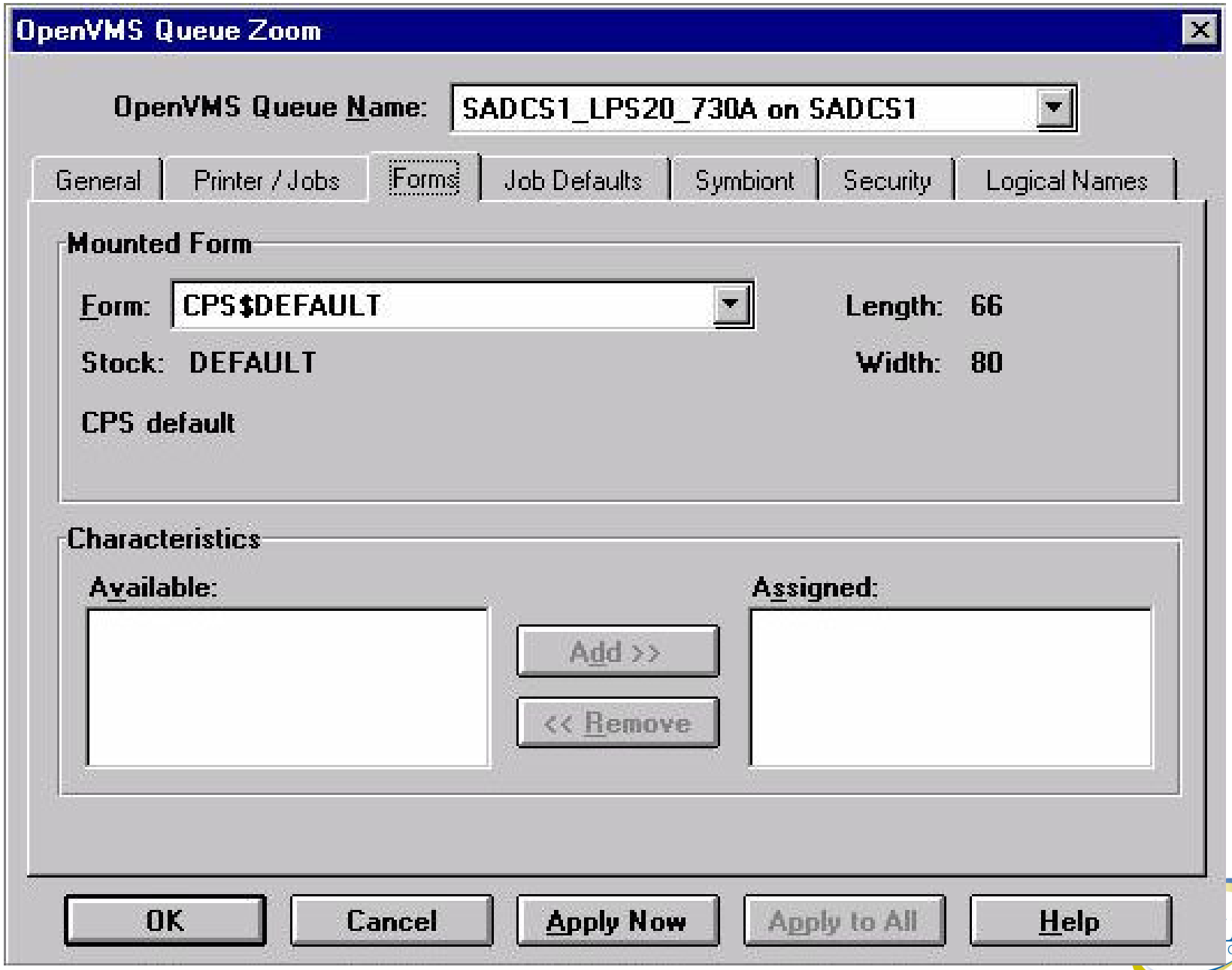

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Privileges
Available
ACNT
ALLSPOOL
ALTPRI
AUDIT
BUGCHK
Grant>>
<< Revoke
Authorized Privileges: (All)
Name
Default
NETMBX
OPER
SETPRV
TMPMBX
Rights Identifiers
System Rights...
Grant>>
<< Revoke
Held Rights
Dynamic
No Access
Resource
Subsystem
OK
Cancel
Apply Now
Apply All
Help

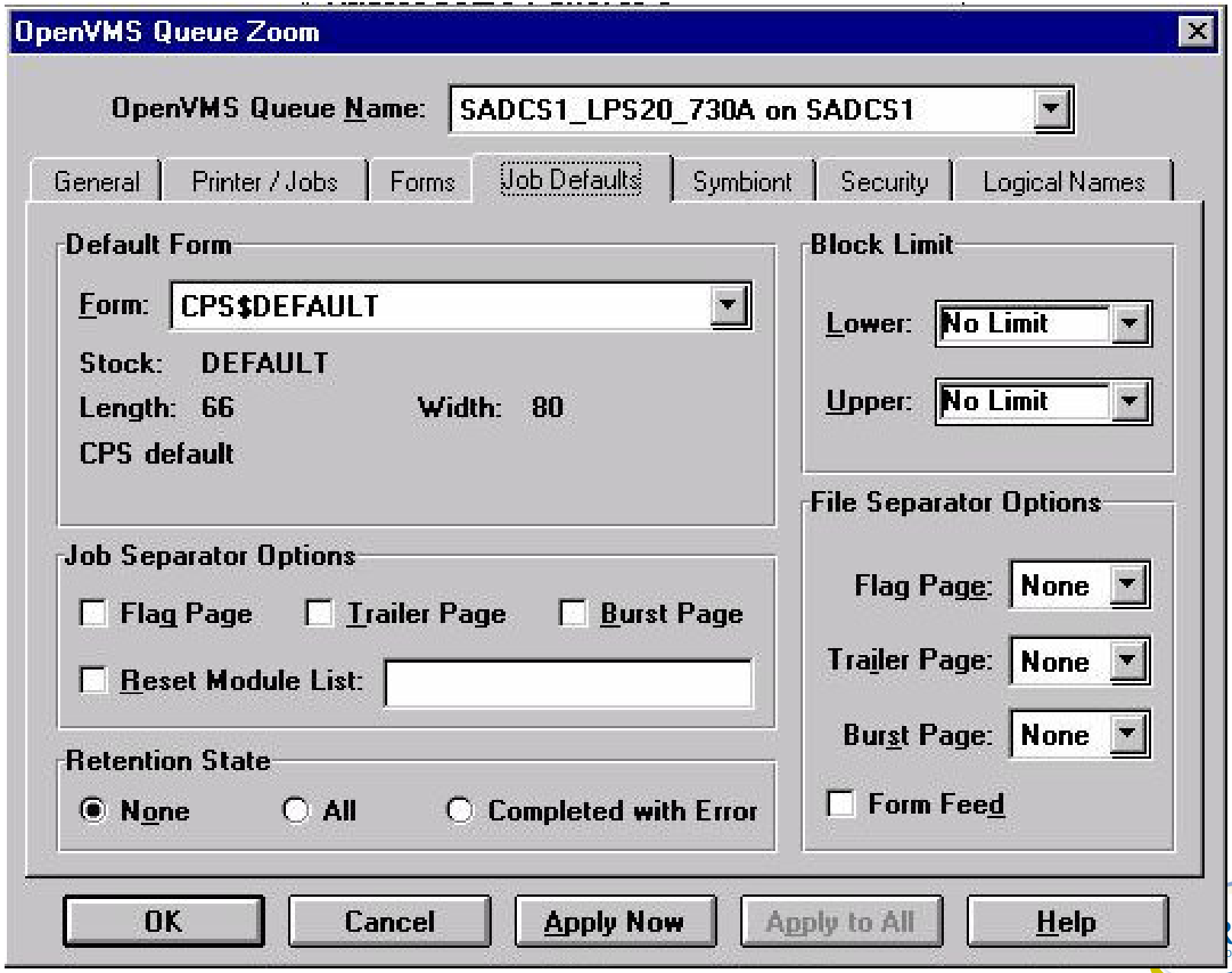

ManageWORKS - OpenVMS User Account Zoom
OpenVMS Account Name:
05369 on SADCSA
General
Flags
Mail
Network Proxies
Passwords
Quotas
Security
Time Restrictions
Primary Days
Monday
Tuesday
Wednesday
Thursday
Friday
Secondary Days
Saturday
Sunday
Time Restrictions
Processing Mode
Local
Batch
Network
Dialup
Remote
Primary Day Access:
Full Access
3 AM
6 AM
9 AM
Noon
3 PM
6 PM
9 PM
Secondary Day Access:
Full Access
3 AM
6 AM
9 AM
Noon
3 PM
6 PM
9 PM
OK
Cancel
Apply Now
Apply All
Help

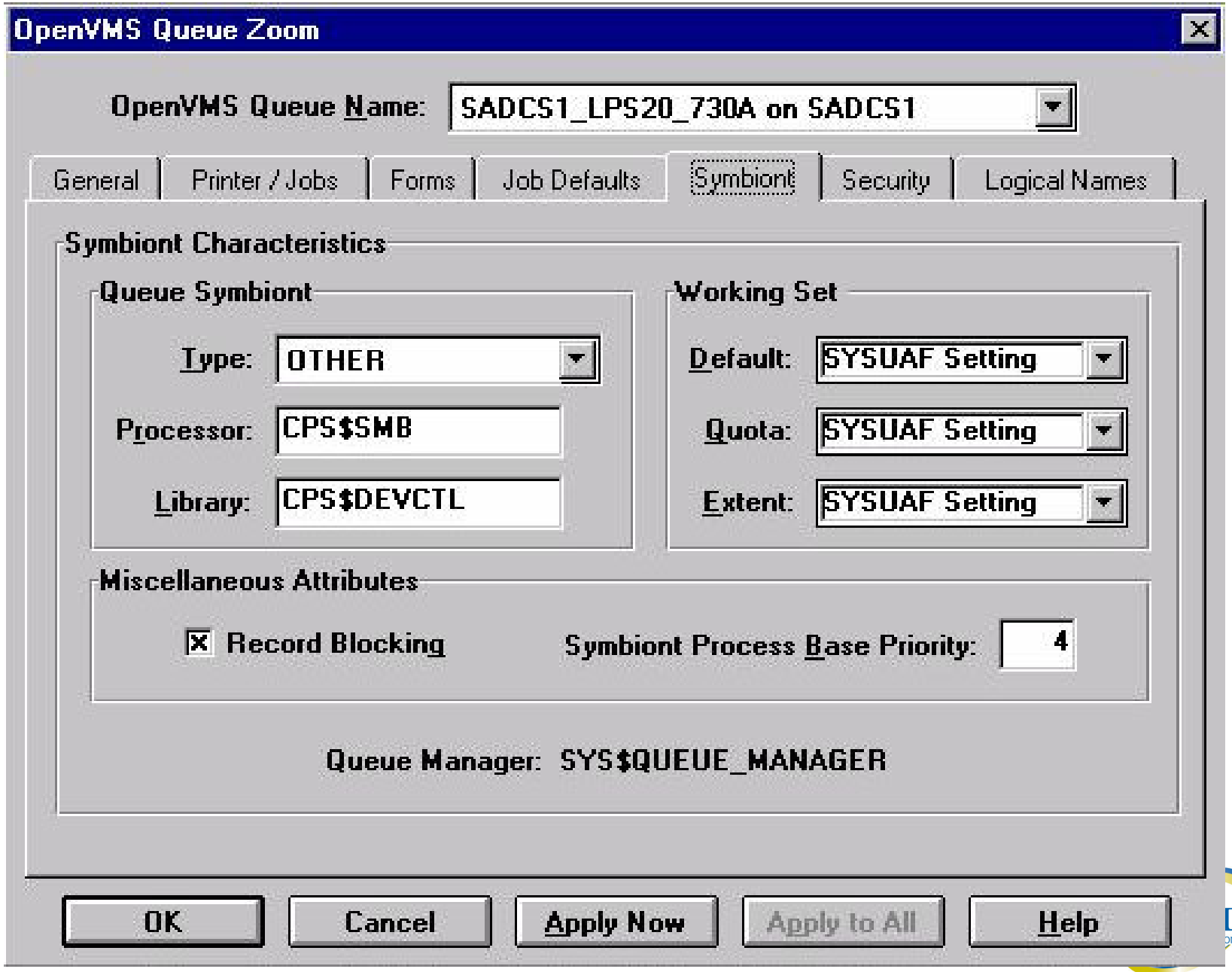

SADCS1
SYSTEM
OpenVMS Accounts
Username Equal to *
OpenVMS Printers
Printer name Equal to *
_LPS20
Started by your command procedures
730B_1216_ADMIN
Started by your command procedures
APPLE_723
Started by your command procedures
APPLE_723A_TINA
Started by OpenVMS Management Station
APPLE_730A_BAKER
Started by OpenVMS Management Station
APPLE2_723
Started by OpenVMS Management Station
DEBBY_LASER
Started by your command procedures
GEORGE
Started by your command procedures
GRIFFIN_QUEUE
Started by OpenVMS Management Station
LPS17_1A
Started by your command procedures
LPS20_730
Started by your command procedures
SADCS1_LPS17
Started by your command procedures
SADCS1_LPS17_730_1A
Started by OpenVMS Management Station
SADCS1_LPS20
Started by your command procedures
SADCS1_LPS20_730A
Started by OpenVMS Management Station
YOUNG
Started by OpenVMS Management Station

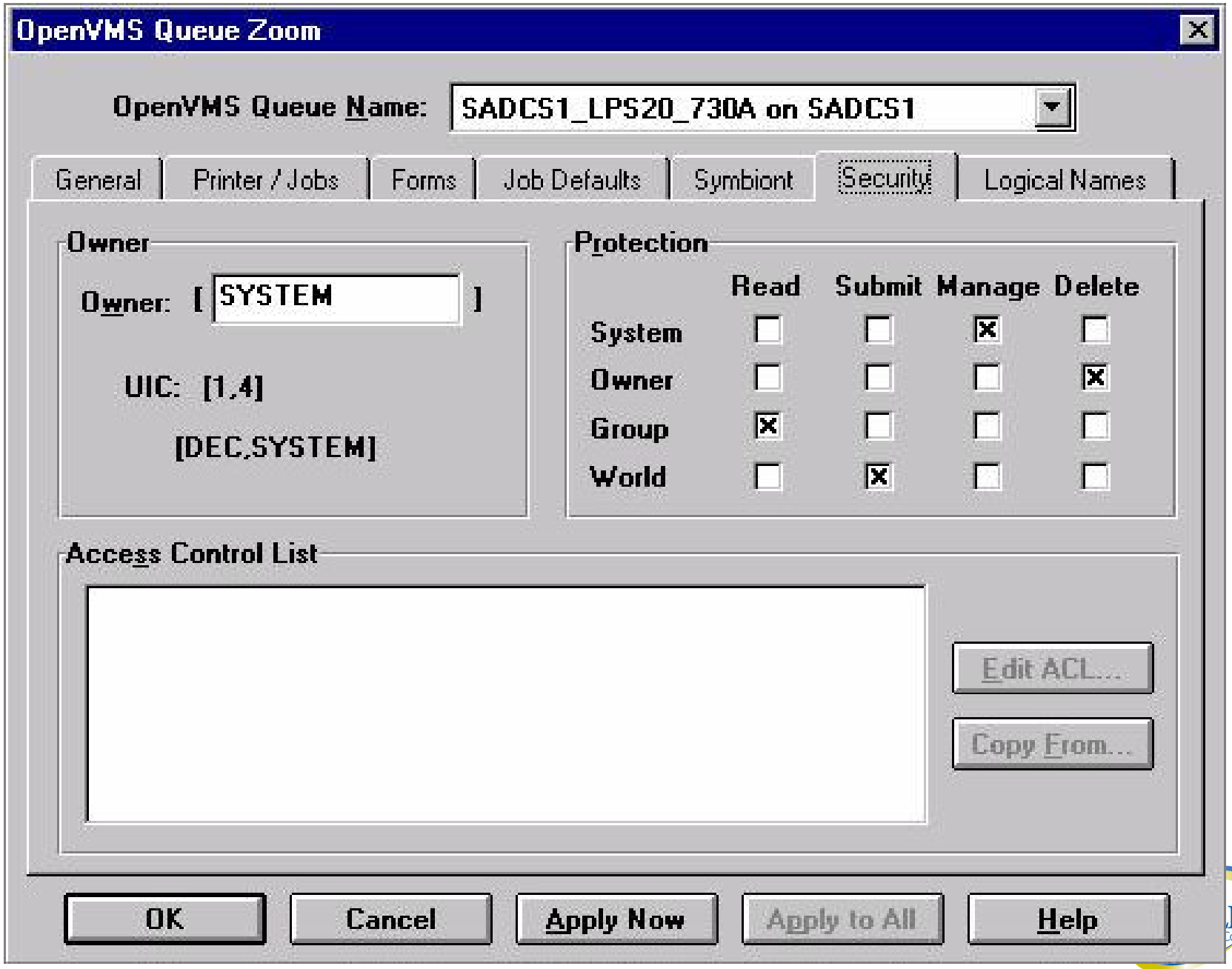

OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
State (Idle)
Started
Stopped
Job Scheduling
Submission Enabled
Schedule by Size
Description:
Execution Queue
Started on: SADCS1::LPS20::
Enable Generic
Failover Nodes...
AutoStart
OK
Cancel
Apply Now
Apply to All
Help

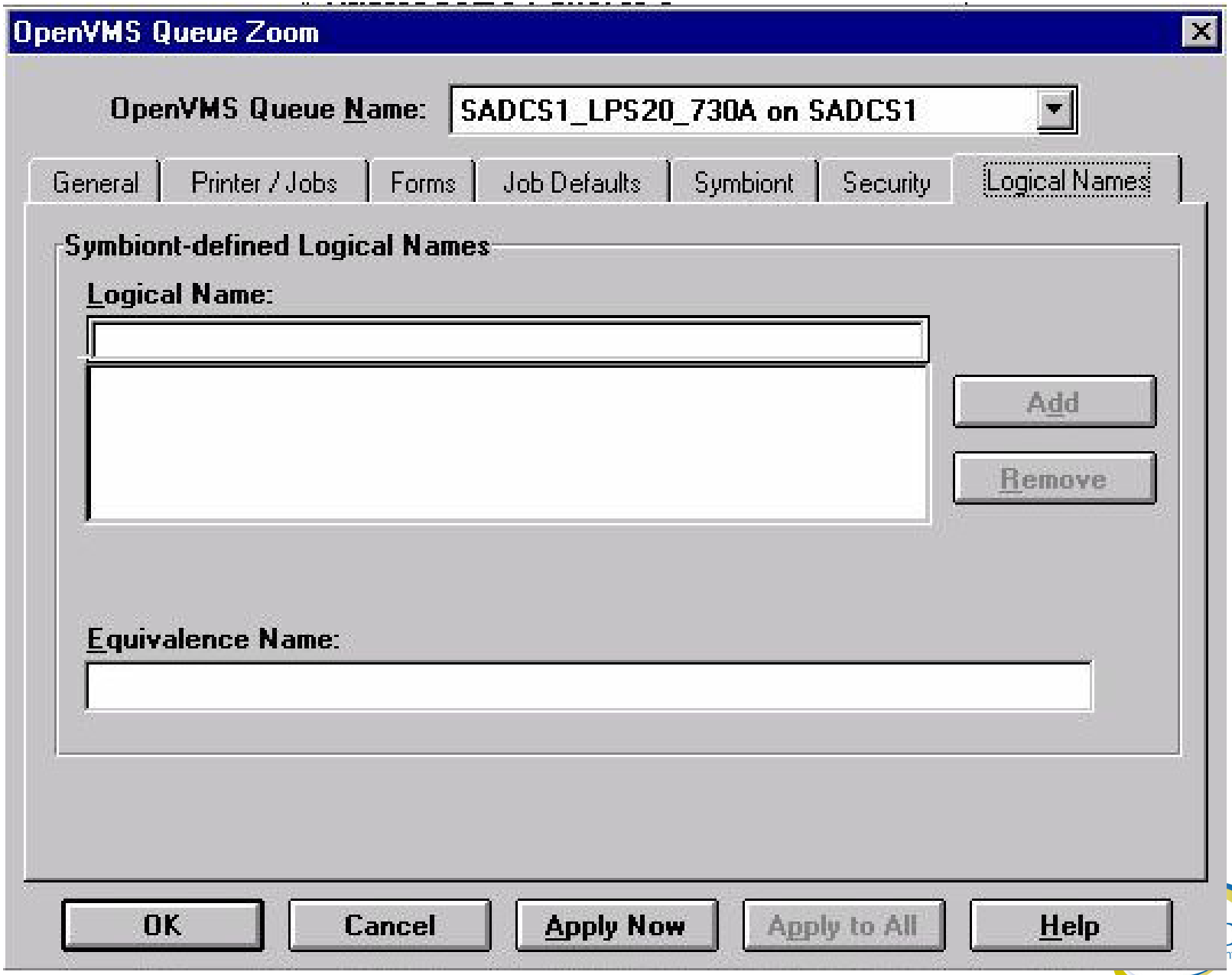

OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
Printer Information
Name: SADCS1_LPS20_730A
Location:
Description:
Jobs:
Job Name
Entry
Status
Owner
JOB
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
Mounted Form
Form:
CPS$DEFAULT
Length: 66
Stock: DEFAULT
Width: 80
CPS default
Characteristics
Available:
Assigned:
Add >>
<< Remove
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
Default Form
Form:
CPS$DEFAULT
Stock: DEFAULT
Length: 66
Width: 80
CPS default
Block Limit
Lower:
No Limit
Upper:
No Limit
File Separator Options
Flag Page:
None
Trailer Page:
None
Burst Page:
None
Form Feed
Job Separator Options
Flag Page
Trailer Page
Burst Page
Reset Module List:
Retention State
None
All
Completed with Error
OK
Cancel
Apply Now
Apply to All
Help

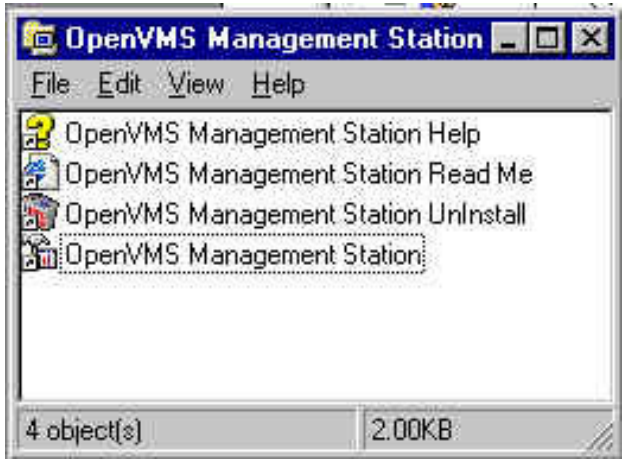

OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
Symbiont Characteristics
Queue Symbiont
Type:
OTHER
Processor:
CPS$SMB
Library:
CPS$DEVCTL
Working Set
Default:
SYSUAF Setting
Quota:
SYSUAF Setting
Extent:
SYSUAF Setting
Miscellaneous Attributes
Record Blocking
Symbiont Process Base Priority:
4
Queue Manager:
SYS$QUEUE_MANAGER
OK
Cancel
Apply Now
Apply to All
Help

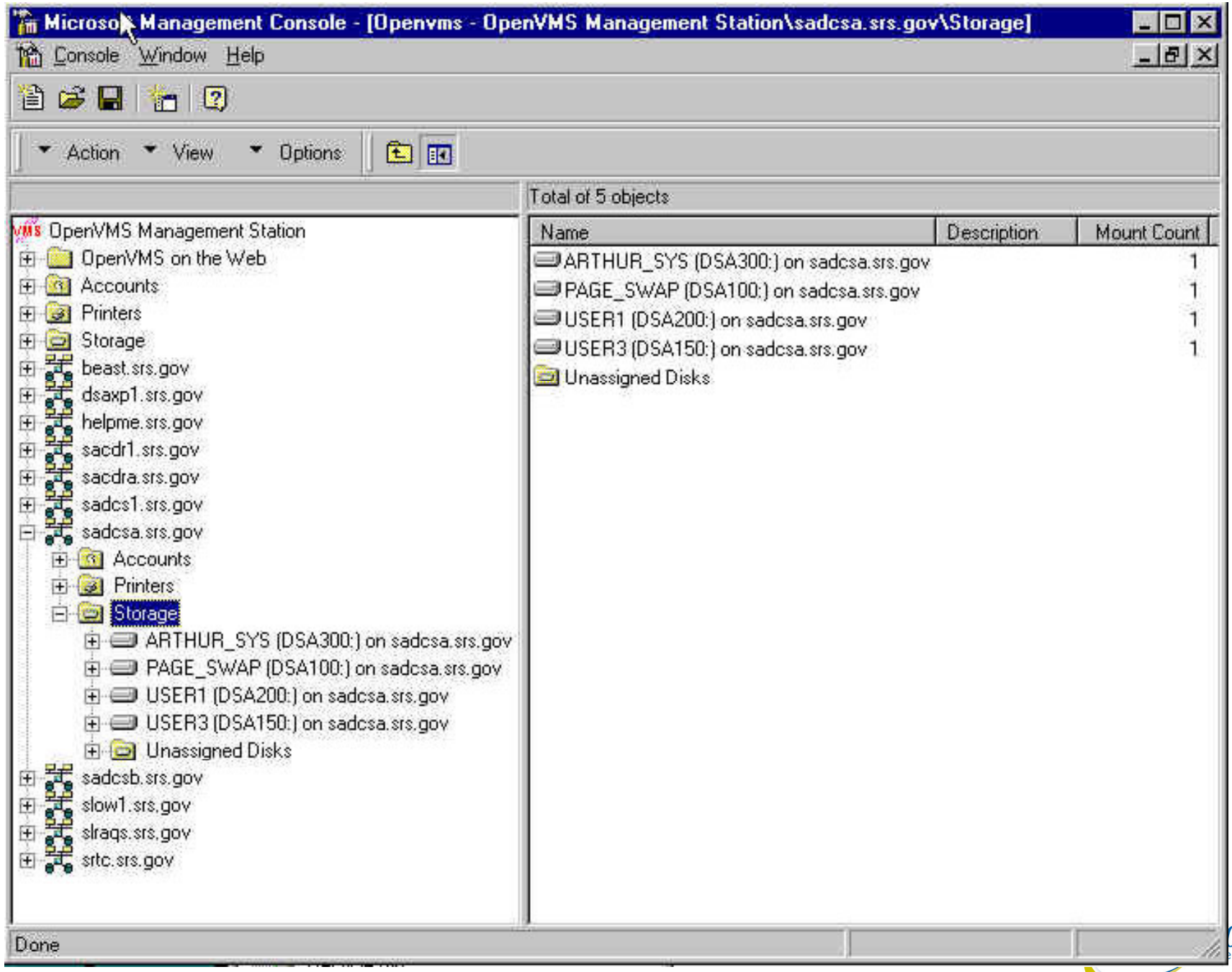

OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
Owner
Owner: [SYSTEM]
UIC: [1,4]
[DEC,SYSTEM]
Protection
Read
Submit
Manage
Delete
System
Owner
Group
World
Access Control List
Edit ACL...
Copy From...
OK
Cancel
Apply Now
Apply to All
Help

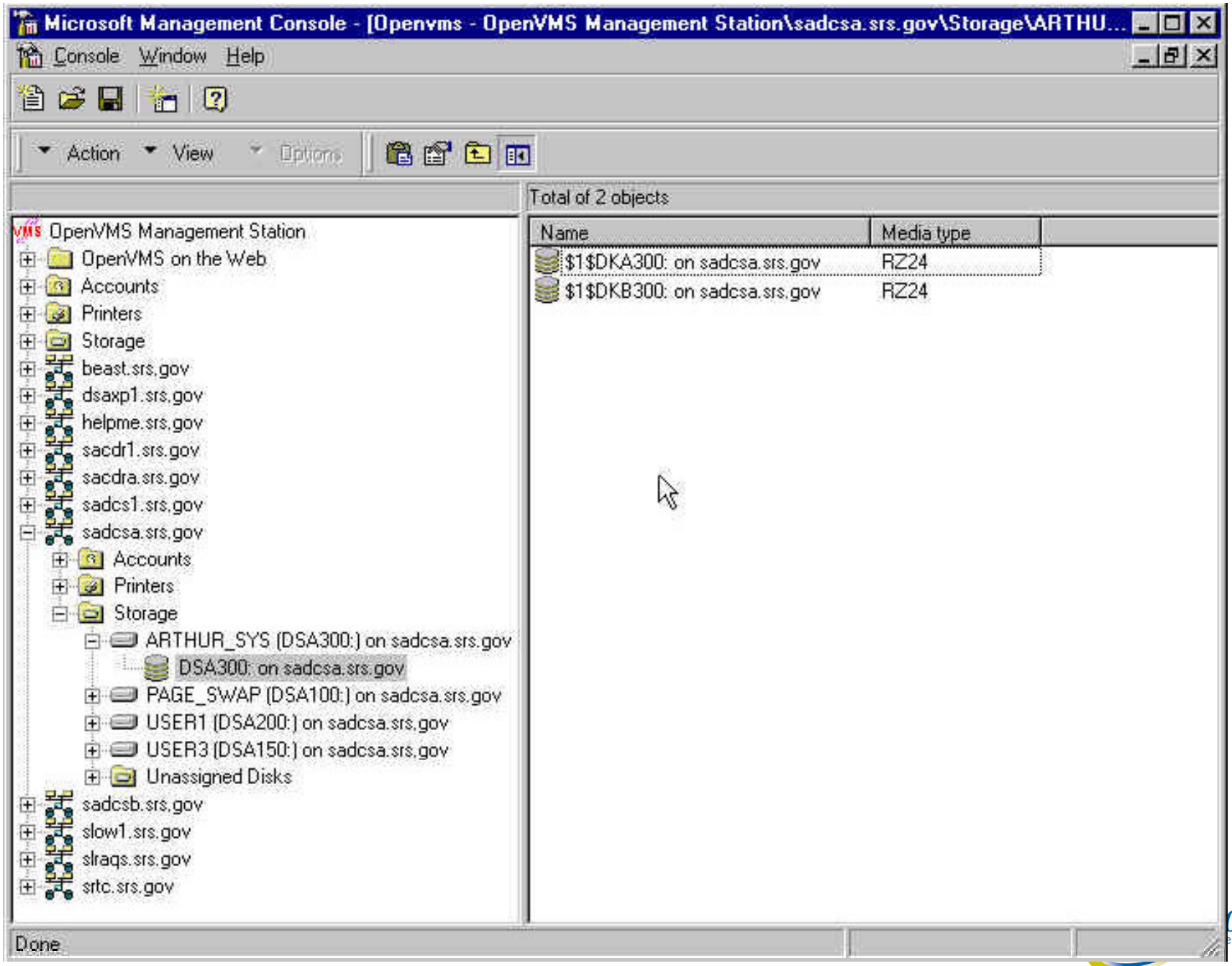

OpenVMS Queue Zoom
OpenVMS Queue Name:
SADCS1_LPS20_730A on SADCS1
General
Printer / Jobs
Forms
Job Defaults
Symbiont
Security
Logical Names
Symbiont-defined Logical Names
Logical Name:
Add
Remove
Equivalence Name:
OK
Cancel
Apply Now
Apply to All
Help

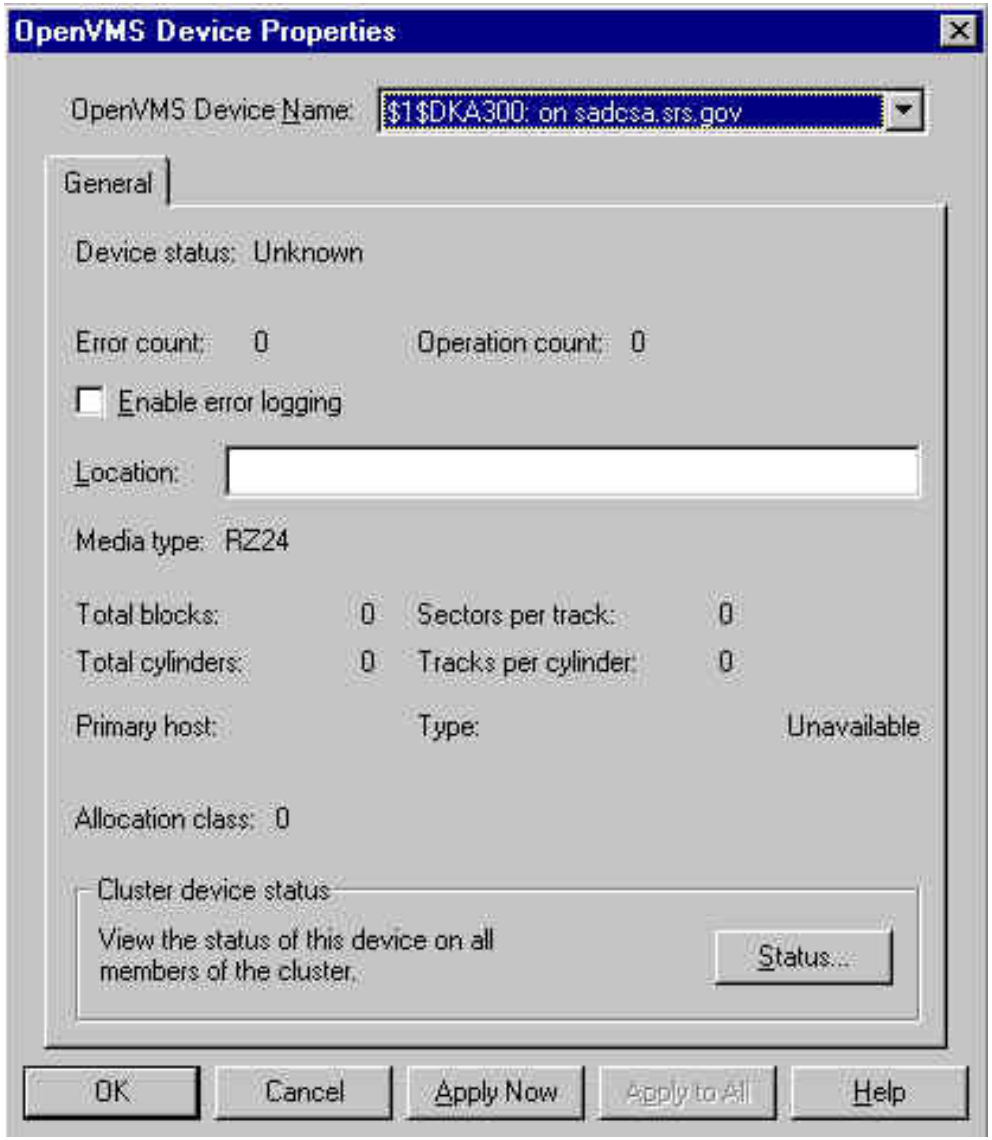
Program Group for Version 3.0

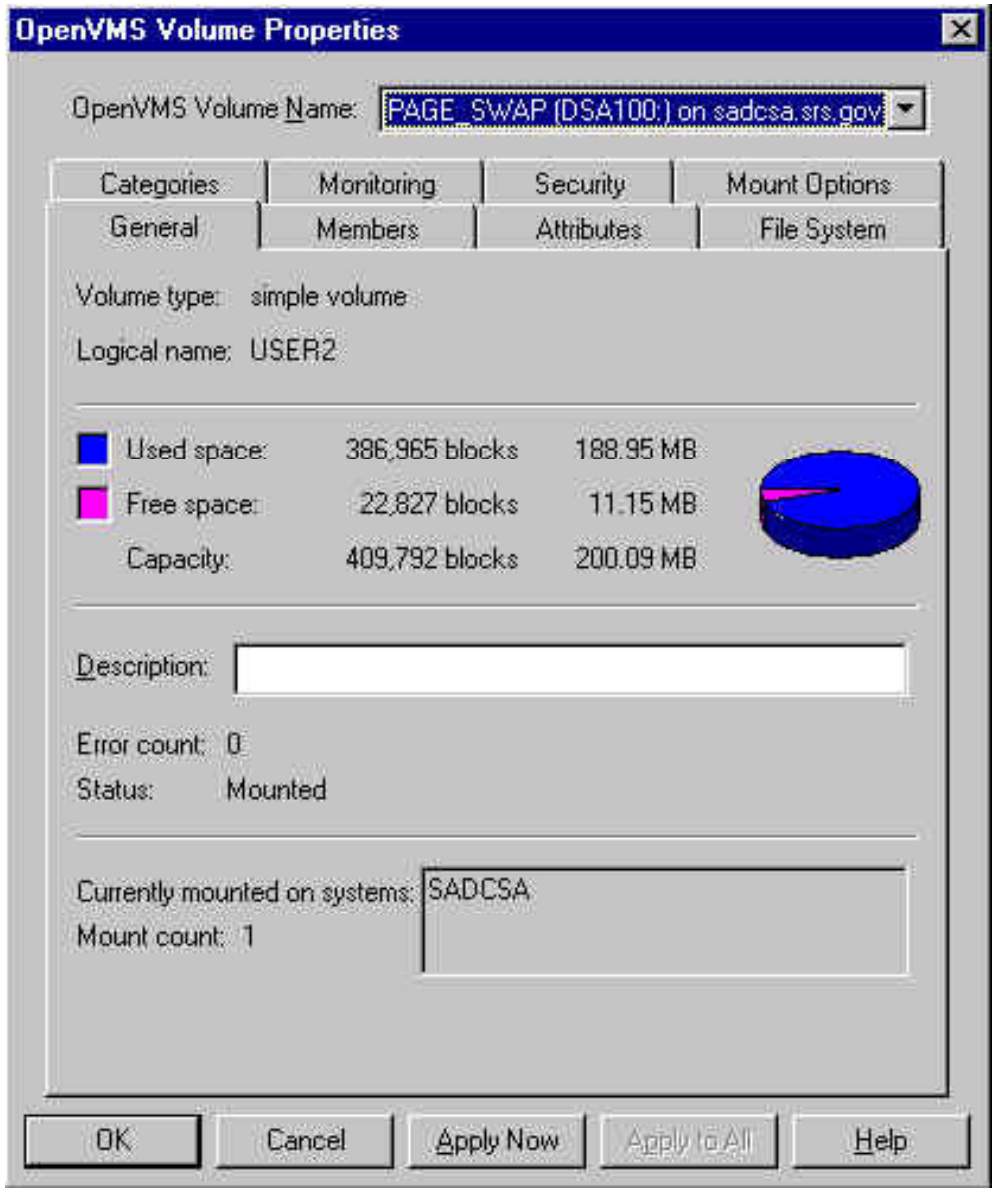
- Program install a different program group
- v2.1 is not uninstalled
- V3.x can communicate to a v2.1 server
  - Just no storage management

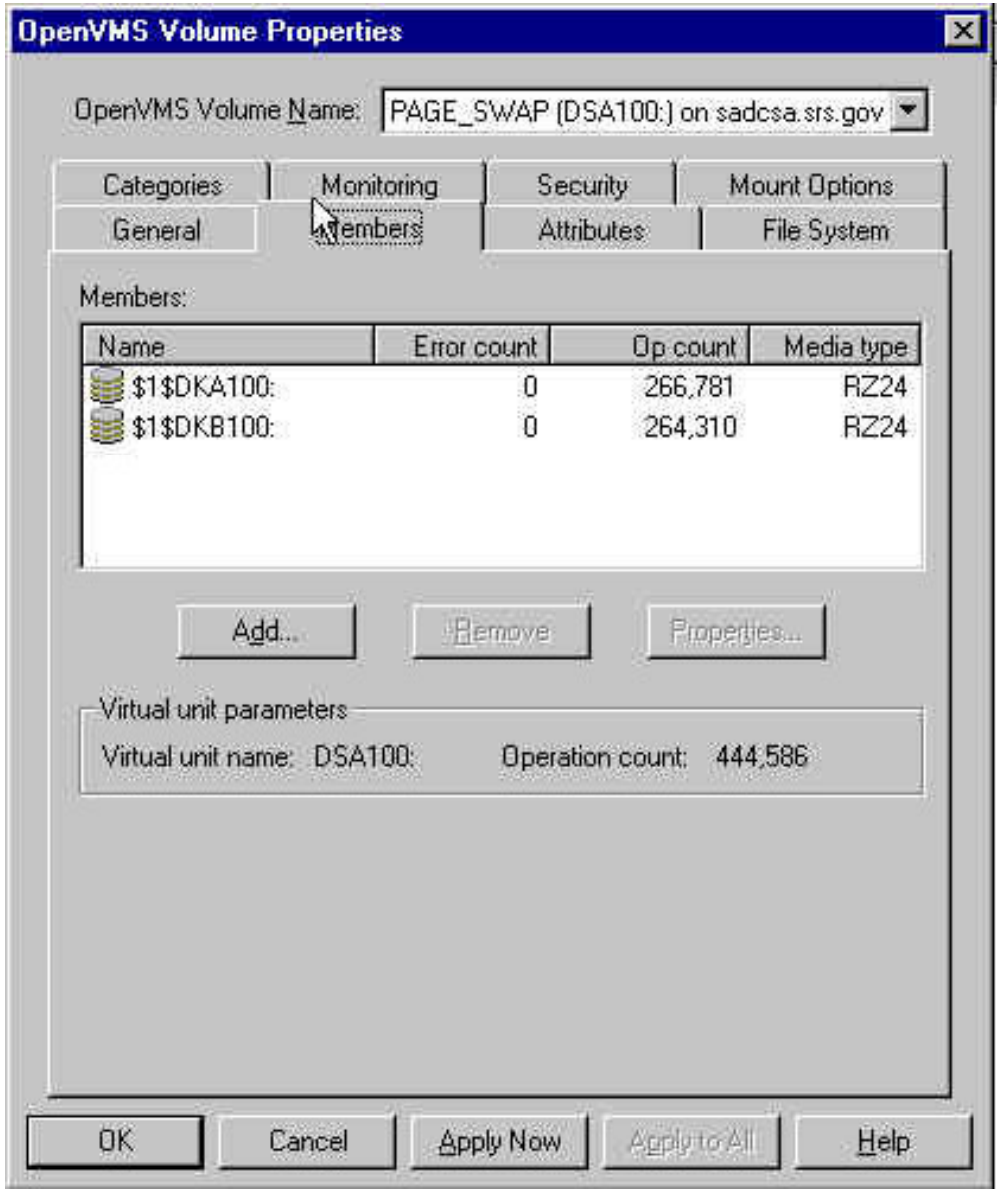


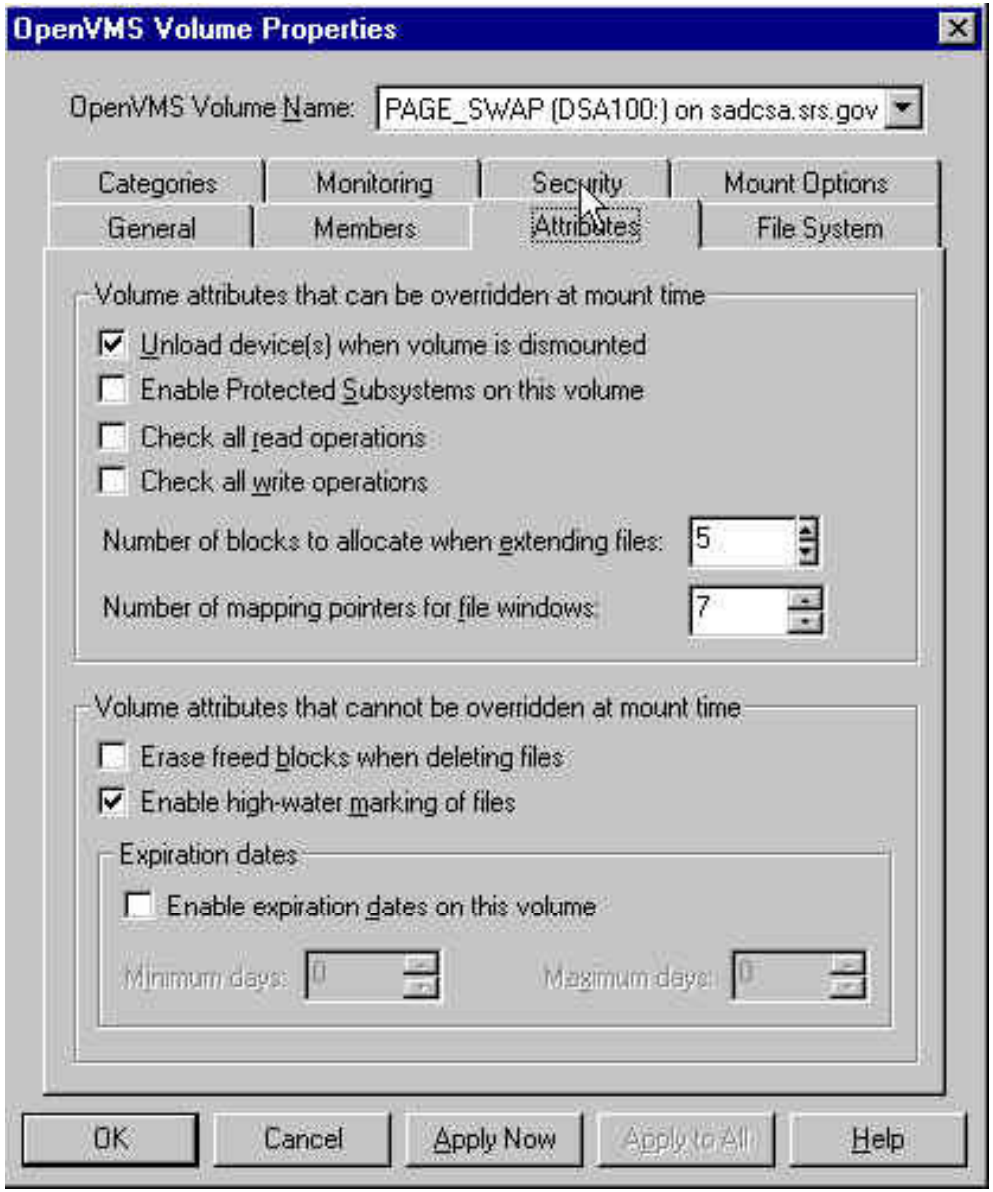

Microsoft Management Console - [Openvms - OpenVMS Management Station\sadcsa.srs.gov\Storage]
Console Window Help
Action View Options
Total of 5 objects
OpenVMS Management Station
OpenVMS on the Web
Accounts
Printers
Storage
beast.srs.gov
dsaxp1.srs.gov
helpme.srs.gov
sacdr1.srs.gov
sacdra.srs.gov
sadcs1.srs.gov
sadcsa.srs.gov
Accounts
Printers
Storage
ARTHUR_SYS (DSA300:) on sadcsa.srs.gov
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
USER1 (DSA200:) on sadcsa.srs.gov
USER3 (DSA150:) on sadcsa.srs.gov
Unassigned Disks
sadcsb.srs.gov
slow1.srs.gov
slraqs.srs.gov
srtc.srs.gov
Name
Description
Mount Count
ARTHUR_SYS (DSA300:) on sadcsa.srs.gov 1
PAGE_SWAP (DSA100:) on sadcsa.srs.gov 1
USER1 (DSA200:) on sadcsa.srs.gov 1
USER3 (DSA150:) on sadcsa.srs.gov 1
Unassigned Disks
Done

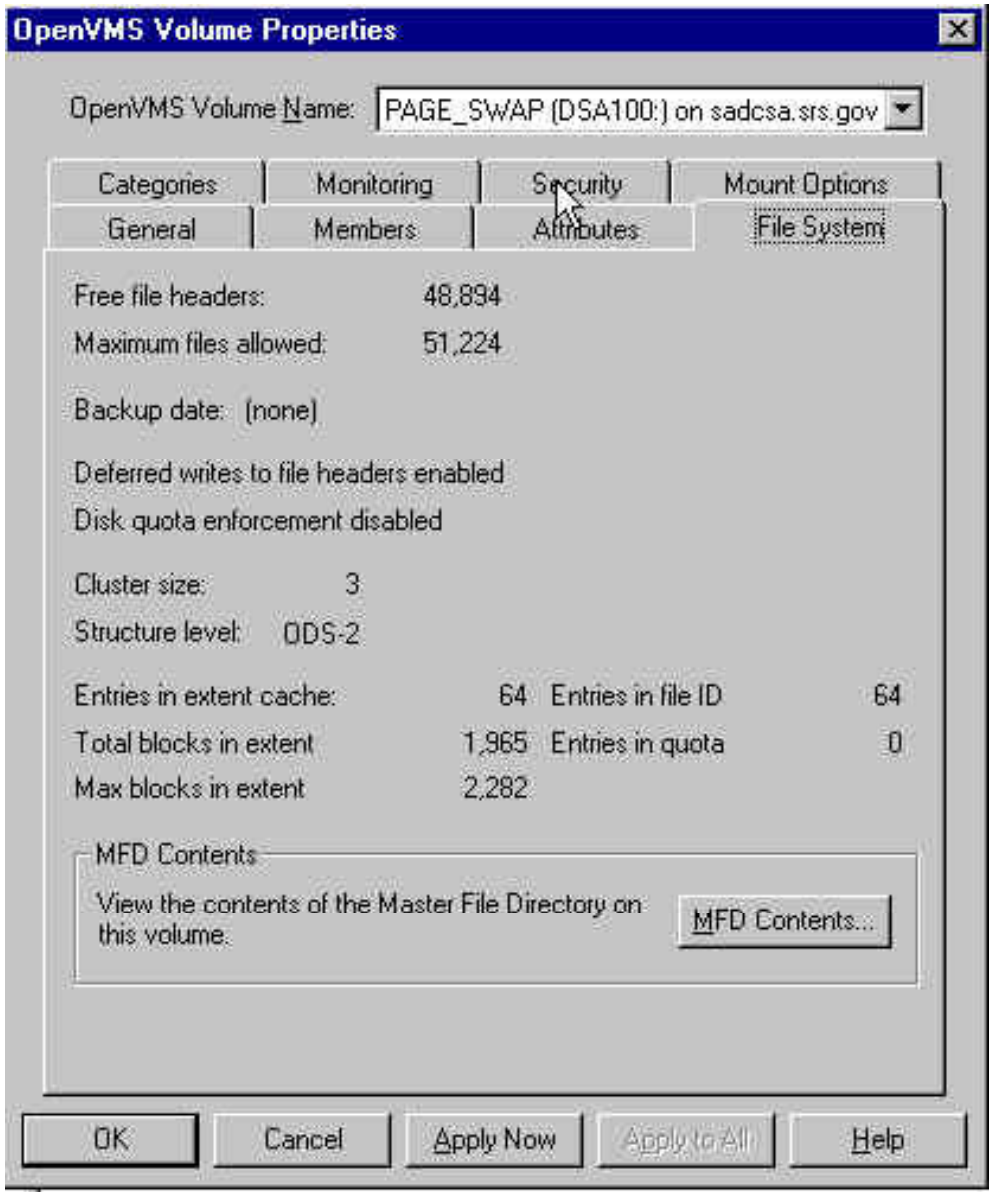

Microsoft Management Console - [Openvms - OpenVMS Management Station\sadcsa.srs.gov\Storage\ARTHU...
Console Window Help
Action View Options
Total of 2 objects
OpenVMS Management Station
OpenVMS on the Web
Accounts
Printers
Storage
beast.srs.gov
dsaxp1.srs.gov
helpme.srs.gov
sacdr1.srs.gov
sacdra.srs.gov
sadcs1.srs.gov
sadcsa.srs.gov
Accounts
Printers
Storage
ARTHUR_SYS (DSA300:) on sadcsa.srs.gov
DSA300: on sadcsa.srs.gov
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
USER1 (DSA200:) on sadcsa.srs.gov
USER3 (DSA150:) on sadcsa.srs.gov
Unassigned Disks
sadcsb.srs.gov
slow1.srs.gov
slraqs.srs.gov
srtc.srs.gov
Name
Media type
1DKA300: on sadcsa.srs.gov
RZ24
1DKB300: on sadcsa.srs.gov
RZ24
Done

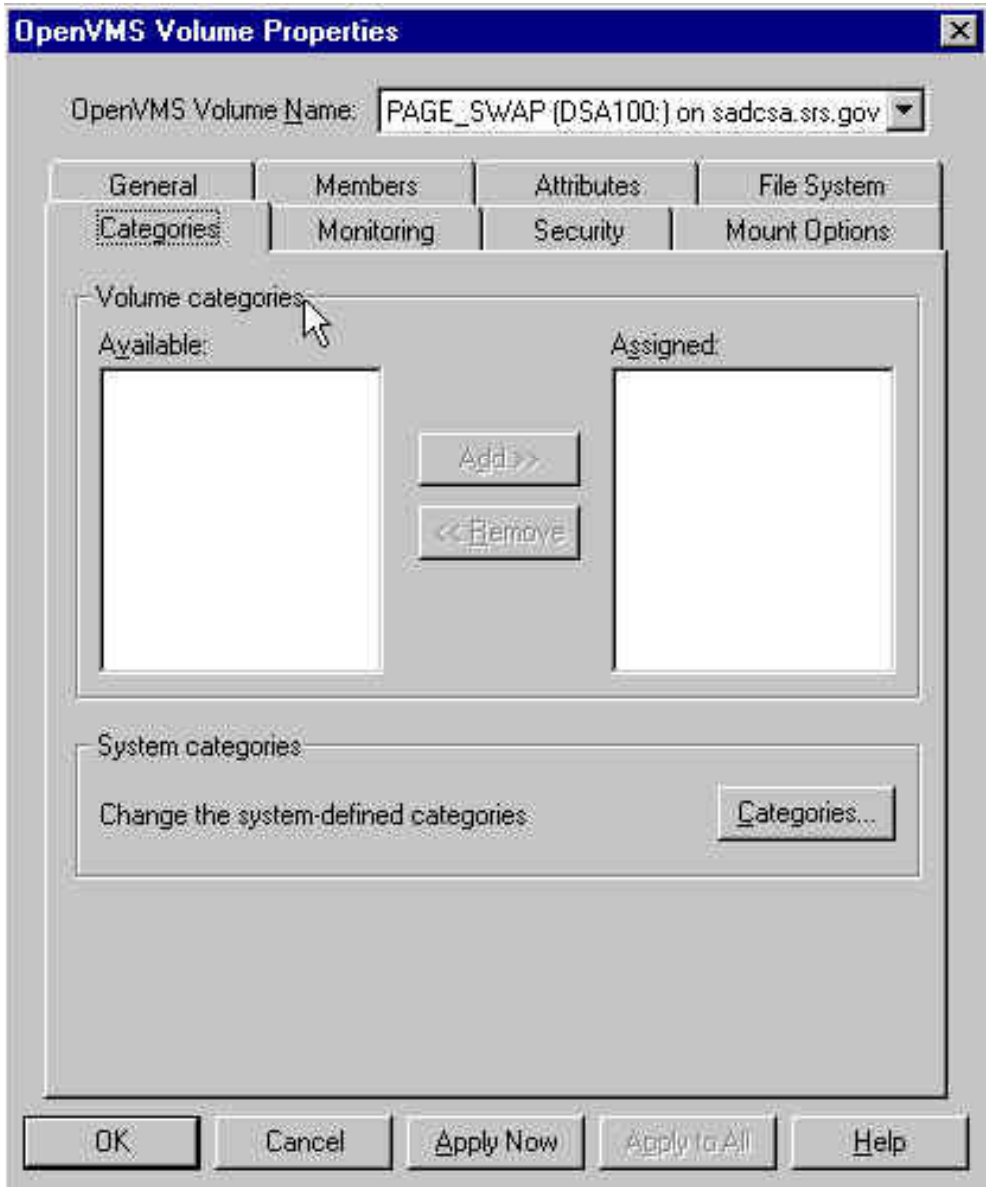

OpenVMS Device Properties
OpenVMS Device Name:
1DKA300: on sadcsa.srs.gov
General
Device status: Unknown
Error count: 0
Operation count: 0
Enable error logging
Location:
Media type: RZ24
Total blocks: 0
Sectors per track: 0
Total cylinders: 0
Tracks per cylinder: 0
Primary host:
Type: Unavailable
Allocation class: 0
Cluster device status
View the status of this device on all members of the cluster.
Status...
OK
Cancel
Apply Now
Apply to All
Help

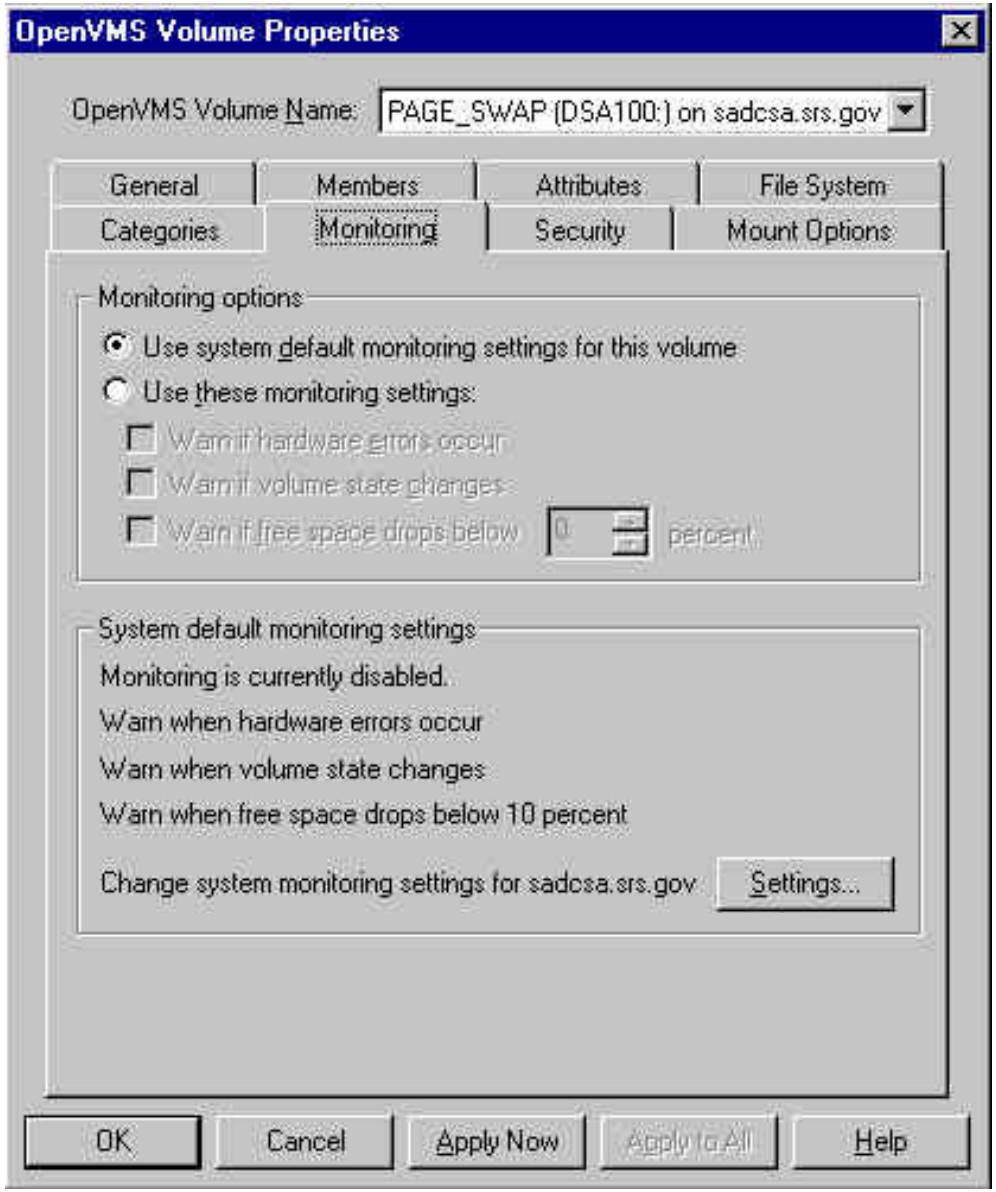

OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
Categories
Monitoring
Security
Mount Options
General
Members
Attributes
File System
Volume type: simple volume
Logical name: USER2
Used space: 386,965 blocks 188.95 MB
Free space: 22,827 blocks 11.15 MB
Capacity: 409,792 blocks 200.09 MB
Description:
Error count: 0
Status: Mounted
Currently mounted on systems:
SADCSA
Mount count: 1
OK
Cancel
Apply Now
Apply to All
Help

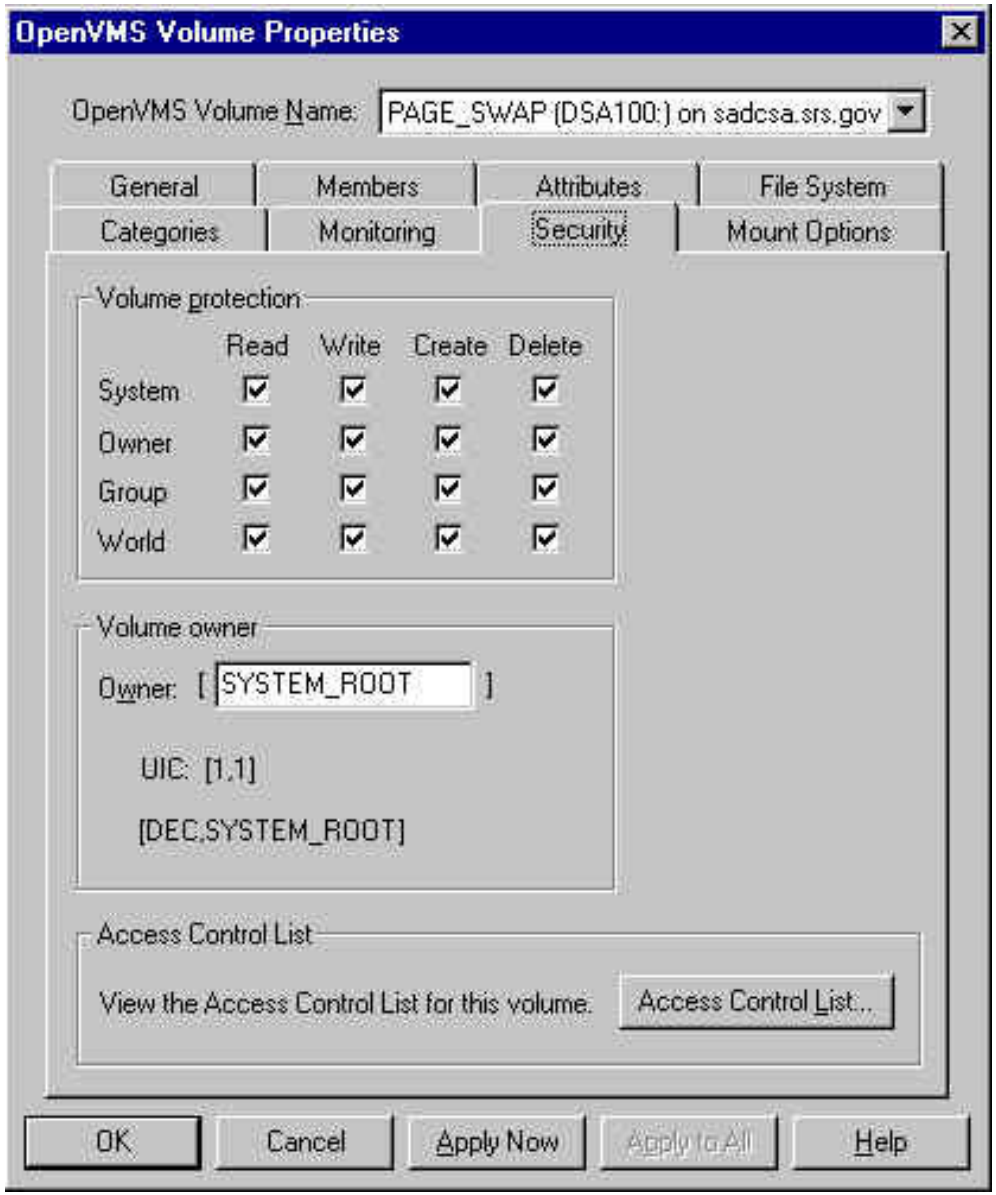

OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
Categories
Monitoring
Security
Mount Options
General
Members
Attributes
File System
Members:
Name
Error count
Op count
Media type
1DKA100:
0
266,781
RZ24
1DKB100:
0
264,310
RZ24
Add...
Remove
Properties...
Virtual unit parameters
Virtual unit name: DSA100:
Operation count: 444,586
OK
Cancel
Apply Now
Apply to All
Help

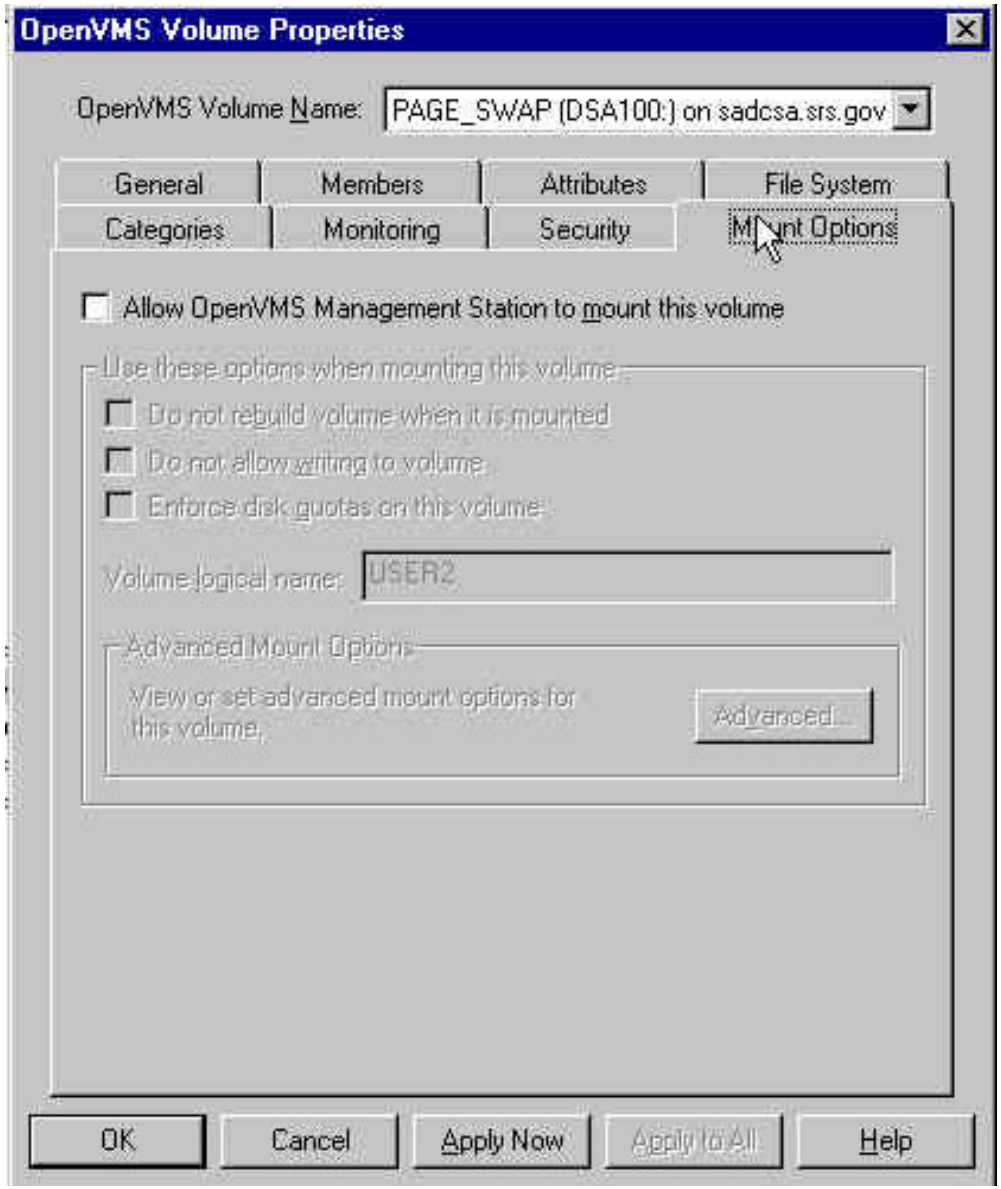

OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
Categories
Monitoring
Security
Mount Options
General
Members
Attributes
File System
Volume attributes that can be overridden at mount time
Unload device(s) when volume is dismounted
Enable Protected Subsystems on this volume
Check all read operations
Check all write operations
Number of blocks to allocate when extending files:
5
Number of mapping pointers for file windows:
7
Volume attributes that cannot be overridden at mount time
Erase freed blocks when deleting files
Enable high-water marking of files
Expiration dates
Enable expiration dates on this volume
Minimum days:
0
Maximum days:
0
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
Categories
Monitoring
Security
Mount Options
General
Members
Attributes
File System
Free file headers: 48,894
Maximum files allowed: 51,224
Backup date: (none)
Deferred writes to file headers enabled
Disk quota enforcement disabled
Cluster size: 3
Structure level: ODS-2
Entries in extent cache: 64
Entries in file ID 64
Total blocks in extent 1,965
Entries in quota 0
Max blocks in extent 2,282
MFD Contents
View the contents of the Master File Directory on this volume.
MFD Contents...
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
General
Members
Attributes
File System
Categories
Monitoring
Security
Mount Options
Volume categories
Available:
Assigned:
Add >>
<< Remove
System categories
Change the system-defined categories
Categories...
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
General
Members
Attributes
File System
Categories
Monitoring
Security
Mount Options
Monitoring options
Use system default monitoring settings for this volume
Use these monitoring settings:
Warn if hardware errors occur
Warn if volume state changes
Warn if free space drops below
percent
System default monitoring settings
Monitoring is currently disabled.
Warn when hardware errors occur
Warn when volume state changes
Warn when free space drops below 10 percent
Change system monitoring settings for sadcsa.srs.gov
Settings...
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
General
Members
Attributes
File System
Categories
Monitoring
Security
Mount Options
Volume protection
Read
Write
Create
Delete
System
Owner
Group
World
Volume owner
Owner: [SYSTEM_ROOT]
UIC: [1,1]
[DEC,SYSTEM_ROOT]
Access Control List
View the Access Control List for this volume.
Access Control List...
OK
Cancel
Apply Now
Apply to All
Help


OpenVMS Volume Properties
OpenVMS Volume Name:
PAGE_SWAP (DSA100:) on sadcsa.srs.gov
General
Members
Attributes
File System
Categories
Monitoring
Security
Mount Options
Allow OpenVMS Management Station to mount this volume
Use these options when mounting this volume:
Do not rebuild volume when it is mounted
Do not allow writing to volume
Enforce disk quotas on this volume
Volume logical name:
USER2
Advanced Mount Options:
View or set advanced mount options for this volume.
Advanced...
OK
Cancel
Apply Now
Apply to All
Help

THE END!!!

- Disclaimer – This information is accurate to the best of my knowledge
- Questions?
- Comments?
- Advise?
- Rotten Tomatoes?
- THANK YOU for attending
- I hope you learned at least one thing to help you
- Fill out an evaluation form

Speaker Contact Information

- F. Arthur Cochrane
- Work email - arthur_cochrane@csx.com
- Other email – cochrane@eisner.encompasserve.org
- Web page - http://eisner.encompasserve.org/~cochrane/
- Office Phone – 904-633-5731

HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

interex

shared knowledge • shared power

encompass

AN HP USER GROUP